# Modeling the Push Mechanism for WGSN

Lin-Yi Wu, Ming-Feng Chang, and Yi-Bing Lin*

***Abstract*—A UMTS and WLAN interworking solution called *WLAN-based GPRS Support Node* (WGSN) has been proposed to allow a UMTS/WLAN dual mode *Mobile Station* (MS) to access heterogeneous wireless services. To reduce the power consumption of an MS, most WGSN applications are not activated at the MS. A push mechanism called *SIP-based Push Center* (SPC) has been implemented in the WGSN node. For an incoming call to an MS that has not turned on WLAN module, the SPC utilizes the UMTS short message service to activate the WLAN access for the MS. This paper studies the performance of SPC. An analytic model is proposed to derive the expected number of lost calls during the activation period. The analytic result is validated against simulation experiments. Our study quantitatively indicates how the SPC performance is affected by the activation time and the timeout period.**

***Index Terms*—*Session Initiation Protocol* (SIP), *Short Message Service* (SMS), *Universal Mobile Telecommunications System* (UMTS), *Wireless LAN* (WLAN).**

## I. Introduction

In [1], we proposed a *Universal Mobile Telecommunications System* (UMTS) and *Wireless LAN* (WLAN) interworking solution called *WLAN-based GPRS Support Node* (WGSN). In the WGSN architecture, the UMTS network (Fig. 1 (1)) provides 3G *Packet Switched* (PS) services, and the WLAN (Fig. 1 (2)) provides access to the Internet. To support mobile roaming between the UMTS network and the WLANs, a *Mobile Station* (MS; see Fig. 1 (3)) must be a 3G-WLAN dual mode terminal equipped with both a WLAN *Network Interface Card* (NIC) and a 3G module.

On the network side, a WGSN node acts as a gateway between the *Packet Data Network* (PDN) and the MSs. To support WGSN mobility management based on the UMTS mechanism, the WGSN node communicates with the *Home Location Register* (HLR) where the subscriber data and the location information of WLAN users are stored. In addition, the UMTS SIM-based authentication is reused for WGSN.

To reduce the power consumption of an MS, most WGSN applications are not activated at the MS (i.e., the WLAN module is turned off) until the user actually accesses them. This approach does not support "always-on" or MS-terminated services, such as incoming *Voice Over IP* (VoIP) calls [2]. To address this issue, a push mechanism called *SIP-based Push Center* (SPC) has been implemented in the WGSN node [1]. In this approach, the mobile *Short Message Service* (SMS) mechanism, which consumes much less power than the WLAN modules, is always on. When a VoIP caller in the external PDN issues a call request to a WGSN MS through *Session Initiation Protocol* (SIP) [3], the request is first sent to the WGSN node (path (a) in Fig. 1). The SPC checks if the SIP *User Agent* (UA) of the called MS is activated. If so, the request is directly forwarded to the called MS (path (d) in Fig. 1). Otherwise, the SPC sends a GSM short message to the MS to activate the corresponding SIP UA (path (b) in Fig. 1). After the SIP UA is activated, the MS informs the SPC (path (c) in Fig. 1), and the call request from the caller is then delivered to the SIP UA following the standard SIP call setup procedure. The implementation details of the software modules and message flows for SPC can be found in [1]. In this paper, we present an accurate analytic model to study the performance of SPC. This analytic model is validated against simulation experiments.

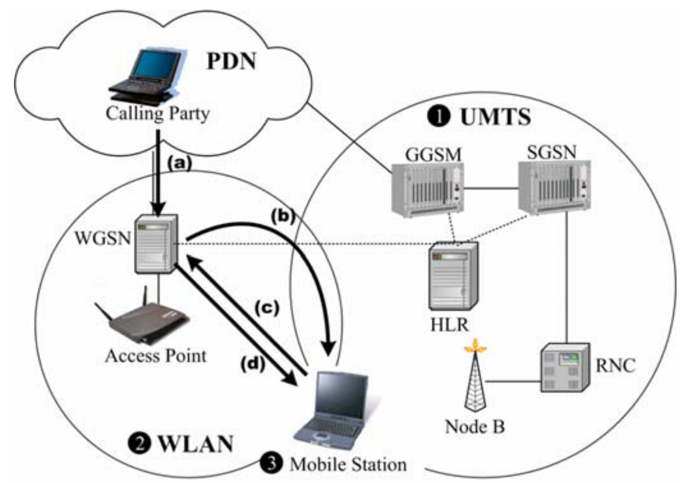



Fig. 1. WGSN Architecture (dashed lines: signaling; solid lines: data and signaling).

## II. The WGSN Push Mechanism

The timing diagram for the execution of SIP UA activation procedure in the SPC mechanism is illustrated in Fig. 2. When the first incoming call arriving at time $\tau_0$ (Fig. 2 (1)), the SPC detects that the WLAN module of the destination MS is turned off. This incoming call is suspended at the SPC. The SPC sends a GSM short message to activate the destination MS (see Fig. 1 (b)) and sets the timer *T1* for this call. The incoming call waiting for setup is referred to as the *outstanding call*. If the activation procedure is not complete before *T1* expires, the call is dropped. In Fig. 2, the timer *T1* for the first outstanding call expires at time $\tau_2$ (Fig. 2 (4)), and the SPC receives the

* Corresponding Author: Yi-Bing Lin, Dept. Comp. Sci. & Info. Engr., National Chiao Tung University, Hsinchu, Taiwan, R.O.C.; Fax:+886-3.5724176; Email: liny@csie.nctu.edu.tw

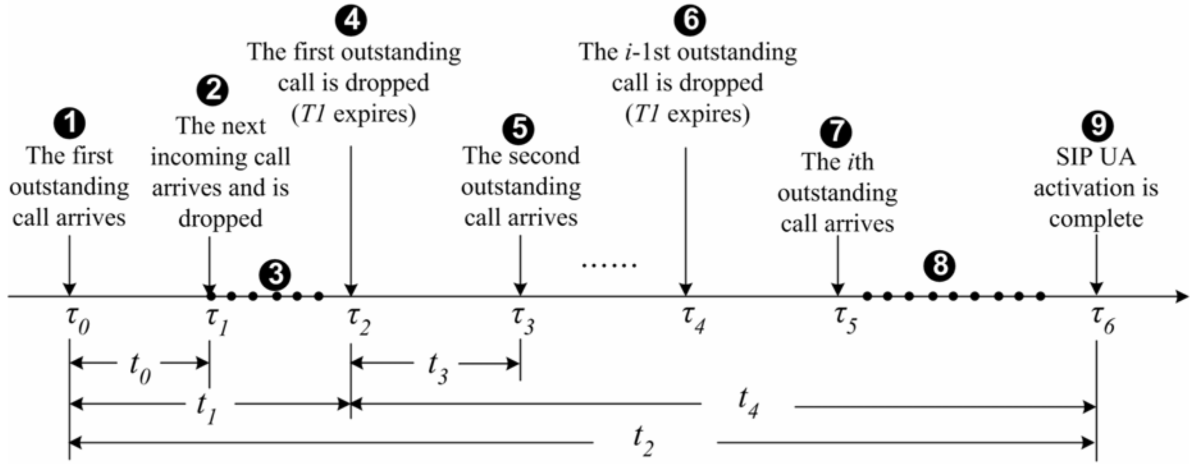



Fig. 2. Timing Diagram for SIP UA Activation (A dot "." represents dropping of an incoming call immediately after it arrives at the SPC)

activation complete message from the called MS at time $\tau_6$ (Fig. 2 (9)), where $\tau_6 > \tau_2$. During SIP UA activation, new incoming calls for the destination MS may arrive. If the outstanding call has not been dropped when a new incoming call arrives, then this new incoming call is dropped (see Fig. 2 (2), (3), and (8)). Otherwise, this incoming call becomes the next outstanding call (see Fig. 2 (5) and (7)).

In this paper, we study the performance of SIP push mechanism, where the expected number of lost calls during SIP UA activation is observed. The lost calls include the dropped outstanding calls due to *T1* expiration (see Fig. 2 (4) and (6)) and the incoming calls arriving when an outstanding call exists (see Fig. 2 (2), (3), and (8)).

## III. Performance Analysis

We propose an analytic model to investigate the performance of SIP push mechanism. The output measure is the expected number $E[N]$ of the lost calls during the activation period. We make the following assumptions:

1) The incoming call arrivals are a Poisson process with rate $\lambda$; therefore, the inter call arrival time $t_0$ is Exponentially distributed with the density function $f_{t_0}(t_0) = \lambda e^{-\lambda t_0}$. In Fig. 2, $t_0 = \tau_1 - \tau_0$.
2) The *T1* timeout period (denoted as $t_1$) has the density function $f_{t_1}(t_1) = \mu e^{-\mu t_1}$. In Fig. 2, $t_1 = \tau_2 - \tau_0$.
3) The SIP UA activation time is denoted as $t_2 = \tau_6 - \tau_0$. We assume that $t_2$ to be Exponentially distributed with the mean $1/\gamma$, and the density function $f_{t_2}(t_2) = \gamma e^{-\gamma t_2}$.

SIP UA activation is initiated upon the arrival of the first outstanding call to a WGSN MS. Consider the second event occurring in the activation period. One of the following three situations may occur.

**Situation 1.** The second event is the completion of SIP UA activation (i.e., $t_2 < t_0$ and $t_2 < t_1$ in Fig. 2).

**Situation 2.** The second event is the subsequent incoming call request for the MS (i.e., $t_0 < t_1$ and $t_0 < t_2$ in Fig. 2).

**Situation 3.** The second event is the expiration of *T1* (i.e., $t_1 < t_0$ and $t_1 < t_2$ in Fig. 2).

Let $P_i$ be the probability that Situation $i$ occurs, and $N_i$ be the expected number of lost calls in the whole activation procedure if Situation $i$ occurs. Probability $P_i$ can be expressed as

$$
\begin{aligned}
P_1 &= \Pr[t_2 < t_0, t_2 < t_1] \\
&= \Pr[t_2 < t_1 < t_0] + \Pr[t_2 < t_0 < t_1] \\
&= \int_{t_2=0}^{\infty} \int_{t_1=t_2}^{\infty} \int_{t_0=t_1}^{\infty} f_{t_2}(t_2) f_{t_1}(t_1) f_{t_0}(t_0)\, dt_0 dt_1 dt_2 \\
&\quad + \int_{t_2=0}^{\infty} \int_{t_0=t_2}^{\infty} \int_{t_1=t_0}^{\infty} f_{t_2}(t_2) f_{t_0}(t_0) f_{t_1}(t_1)\, dt_1 dt_0 dt_2 \\
&= \frac{\gamma}{\lambda + \mu + \gamma}
\end{aligned}
$$

$$
\begin{aligned}
P_2 &= \Pr[t_0 < t_1, t_0 < t_2] = \Pr[t_0 < t_1 < t_2] + \Pr[t_0 < t_2 < t_1] \\
&= \frac{\lambda}{\lambda + \mu + \gamma}
\end{aligned}
$$

$$
\begin{aligned}
P_3 &= \Pr[t_1 < t_0, t_1 < t_2] = \Pr[t_1 < t_0 < t_2] + \Pr[t_1 < t_2 < t_0] \\
&= \frac{\mu}{\lambda + \mu + \gamma}
\end{aligned}
\tag{1}
$$

In Situation 1, the first outstanding call is set up after the completion of the SIP UA activation, and no call is lost. Therefore,

$$N_1 = 0 \tag{2}$$

In Situation 2, the second event is the arrival of a new incoming call. This new incoming call is dropped because there already exists one outstanding call. Since $t_1$, $t_2$, and $t_3$ are Exponentially distributed, their residual times are also Exponentially distributed due to the memoryless property [4], and there will be $E[N]$ lost calls after the second event. Therefore, we have

$N_2 = 1 + E[N]$ (3)

In Situation 3, the second event is *T1* timer expiration, which leads to the dropping of the first outstanding call. Consider the expected number of lost calls after *T1* expires. We further categorize this situation into two sub-cases in terms of the third event.

**Situation 3-1.** The third event is the completion of SIP UA activation (i.e., $t_4 < t_3$ in Fig. 2). In this sub-case, no call is lost after the second event.

**Situation 3-2.** The third event is a new incoming call request (i.e., $t_3 < t_4$ in Fig. 2). This incoming call becomes the outstanding call. From the residual life theorem and the memoryless property of the Exponential distribution [4], $t_3$ has the same distribution as that for the inter call arrival time $t_0$. That is, $f_{t_3}(t) = f_{t_0}(t) = \lambda e^{-\lambda t}$. Similarly, we have $f_{t_4}(t) = f_{t_2}(t) = \gamma e^{-\gamma t}$. Since $t_3$ and $t_4$ have the same distributions as those for $t_0$ and $t_2$, respectively, the situation seen by this outstanding call is the same as that seen by the first outstanding call. Therefore, the expected number of lost calls after the arrival of this new outstanding call is $E[N]$.

Let $P_{3\text{-}i}$ denote the probability that Situation 3-i occurs, we have

$$\begin{aligned} N_3 &= 1 + (P_{3-1} \times 0 + P_{3-2} \times E[N]) \\ &= 1 + \Pr[t_3 < t_4] \times E[N] \\ &= 1 + \left[\int_{t_3=0}^{\infty}\int_{t_4=t_3}^{\infty} f_{t_3}(t_3)\, f_{t_4}(t_4)\, dt_4 dt_3\right] \times E[N] \\ &= 1 + \frac{\lambda E[N]}{\lambda + \gamma} \end{aligned} \tag{4}$$

From Equations (1), (2), (3), and (4), we have

$$\begin{aligned} E[N] &= \sum_{i=1}^{3} P_i \times N_i \\ &= \left(\frac{\gamma}{\lambda+\mu+\gamma}\right) \times 0 + \left(\frac{\lambda}{\lambda+\mu+\gamma}\right) \times \left(1 + E[N]\right) \\ &\quad + \left(\frac{\mu}{\lambda+\mu+\gamma}\right) \times \left(1 + \frac{\lambda E[N]}{\lambda+\gamma}\right) \end{aligned} \tag{5}$$

By re-arranging Equation (5), we have

$$E[N] = \frac{(\lambda+\gamma)(\lambda+\mu)}{\gamma(\lambda+\mu+\gamma)} \tag{6}$$

Equations (6) have been validated against the simulation experiments. The flowchart and the results of simulation are described in section 4.

## IV. THE SIMULATION MODEL

We utilize discrete event simulation experiments to validate the analytic model described in Section 3. In the simulation, three types of events are defined: **CallArrival** event represents the arrival of incoming call to an MS, **TimerExpiration** event represents that *T1* of an outstanding call expires, and **ActivationComplete** event indicates that the SIP UA of the called MS is successfully activated. Every event is associated with a timestamp representing when the event occurs. The type of an event *e* is denoted as *e*.type, and the timestamp of *e* is denoted as *e*.timestamp.

In the simulation, all events are first inserted into an event list. Then they are deleted from the event list and processed in the non-decreasing timestamp order. A simulation clock *clk* is maintained to indicate the progress of the simulation. The clock value is the timestamp of the event being processed. The inter call arrival times are produced from an Exponential random number generator with mean $1/\lambda$. The SIP UA activation times are drawn from an Exponential random number generator with mean $1/\gamma$. The *T1* timeout periods are drawn from an Exponential random number generator with mean $1/\mu$. The flag **OutstandingCall** is used in the simulation to indicate if there exists an outstanding call.

To ensure that the simulation results are stable, the SIP UA activation procedure is processed $K=10^7$ iterations. At the *i*th iteration ($i \leqslant K$), the number $n_i$ of lost calls during the SIP UA activation procedure is computed. Finally, the expected number $E[N]$ of lost calls is computed as

$$E[N] = \frac{\sum_{i=1}^{K} n_i}{K} \tag{7}$$

The flowchart of the simulation is shown in Fig. 3, and the details are described as follows.

Step 1. The simulation is initialized (i.e., $K=10^7$, $i=0$).

Step 2. Check if *K* simulation iterations have been complete. If so, go to Step 11. If not, go to Step 3.

Step 3. The *i*th iteration is initialized. Specifically, the event list is reset, *clk* is set as 0, and the number of lost calls $n_i$ is initialized to 0. Since the activation procedure is triggered by the first outstanding call, initially, **OutstandingCall**=*TRUE*. Then, three events are generated and inserted into the event list. The **ActivationComplete** event indicates the completion of the *i*th interation. The **TimerExpiration** event shows that the *T1* for the first outstanding call expires. The **CallArrival** event presents the arrival of second incoming call.

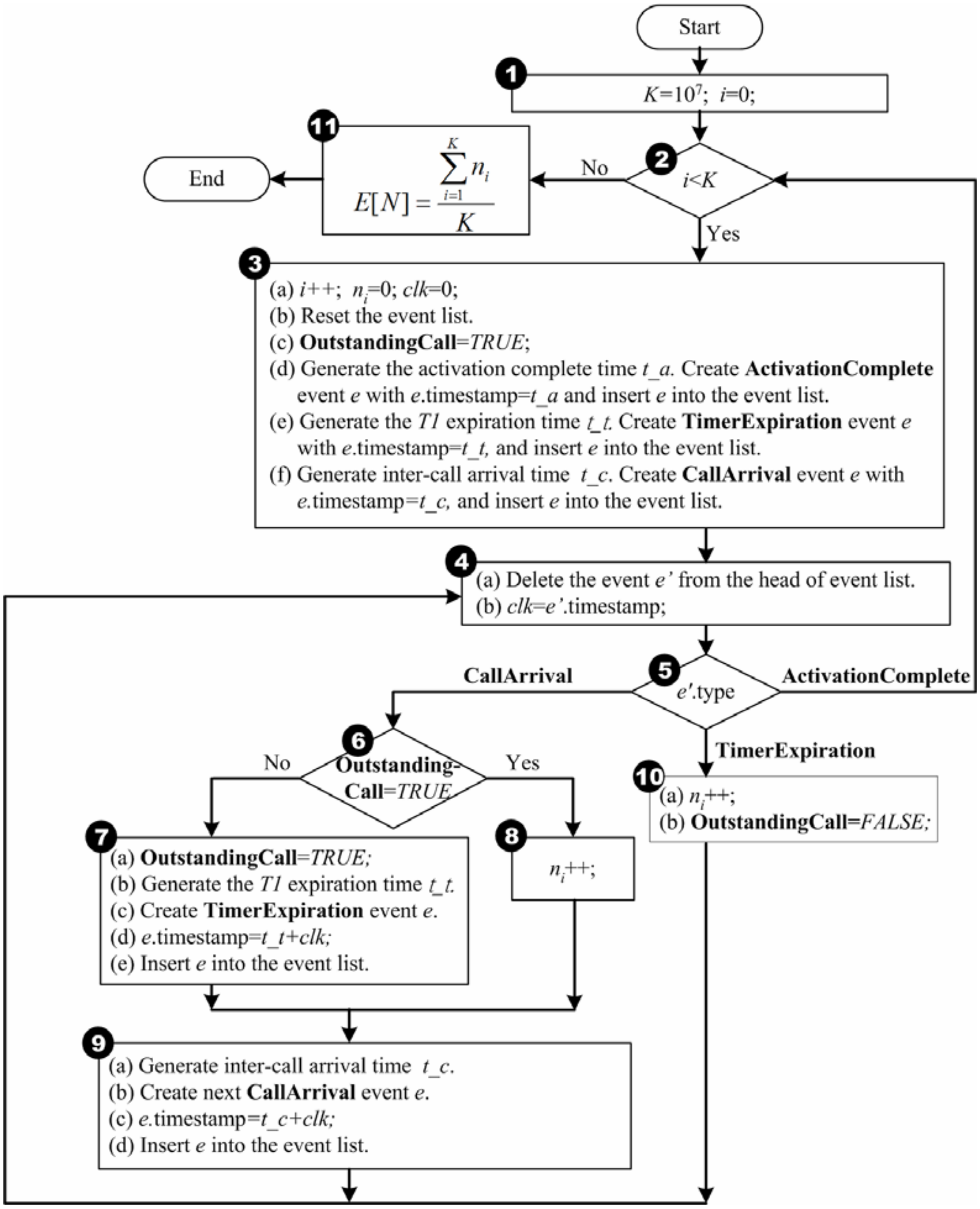



Fig. 3. The Simulation Flowchart

Steps 4 and 5. The next event *e'* is deleted from the head of the event list. The simulation clock *clk*=*e'*.timestamp. If *e'*.type=**CallArrival**, the simulation flow proceeds to Step 6. If *e'*.type=**TimerExpiration**, Step 10 is executed. If *e'*.type=**ActivationComplete**, the SIP UA of the called MS is successfully activated, and the flow goes to Step 2 to initiate another simulation iteration.

Step 6. If **OutstandingCall**=*TRUE* (i.e., there is no outstanding call in the system), go to Step 8. Otherwise, go to Step 7.

Step 7. The new incoming call becomes an outstanding call, and the flag **OutstandingCall** is set to *TRUE*. Then we generate a **TimerExpiration** event *e* to indicate the *T1* expiration for the new outstanding call. *e*.timestamp is set to *t_t*+*clk* where *t_t* is the *T1* expiration time. Finally, *e* is inserted into the event list.

Step 8. If **OutstandingCall**=*TRUE* at Step 6 (i.e., there exists an outstanding call in the system), the new incoming call is dropped. $n_i$ is incremented by 1.

Step 9. Generate the next incoming call event *e*. Specifically, the inter-call arrival time *t_c* is generated, and *e*.timestamp=*t_c*+*clk*. The event type of *e* is **CallArrival**. Finally, *e* is inserted into the event list.

Step 10. If *e'*.type=**TimerExpiration** at Step 5, it means that the outstanding call is dropped. Therefore, **OutstandingCall** is set to *FALSE*, and $n_i$ is incremented by 1.

Step 11. If K iterations of the simulation have been complete, $E[N]$ is computed by using (7), and the simulation is terminated.

Some simulation results are shown in Table I and II. In all scenarios we considered, the discrepancies between the analytic model and simulation are small (less than 0.5% in all cases considered in the tables). Therefore, the accuracy of analytic model is proved.

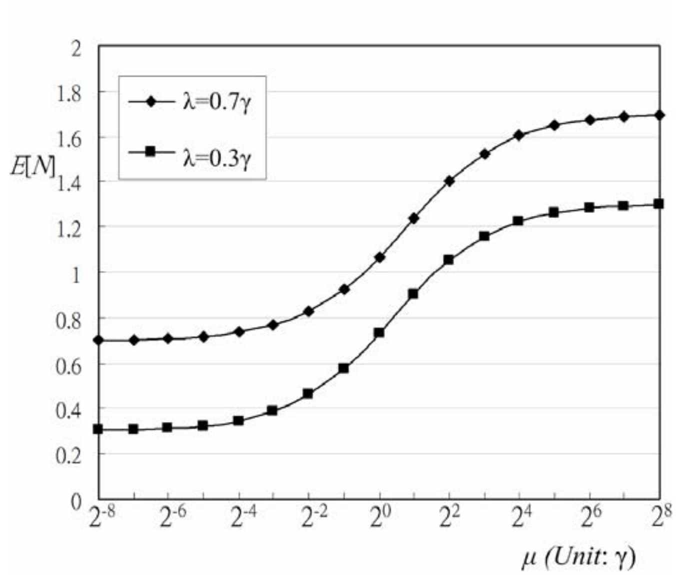
TABLE I
THE $E[N]$ VALUES (ANALYTIC VS. SIMULATION; $\lambda$=0.7 $\gamma$ AND 0.3 $\gamma$)

| $\lambda$=0.7 $\gamma$ | | | | $\lambda$=0.3 $\gamma$ | | | |
|---|---|---|---|---|---|---|---|
| $\mu/\gamma$ | Analytic | Simulation | Errors (%) | $\mu/\gamma$ | Analytic | Simulation | Errors (%) |
| $2^{-8}$ | 0.702293 | 0.70075 | -0.2196 | $2^{-8}$ | 0.302996 | 0.302754 | -0.0798 |
| $2^{-7}$ | 0.704575 | 0.703733 | -0.1194 | $2^{-7}$ | 0.305974 | 0.306641 | 0.2181 |
| $2^{-6}$ | 0.709107 | 0.710396 | 0.1817 | $2^{-6}$ | 0.311876 | 0.311054 | -0.2637 |
| $2^{-5}$ | 0.718051 | 0.718101 | 0.0070 | $2^{-5}$ | 0.323474 | 0.323254 | -0.0681 |
| $2^{-4}$ | 0.735461 | 0.734415 | -0.1422 | $2^{-4}$ | 0.345872 | 0.345896 | 0.0071 |
| $2^{-3}$ | 0.768493 | 0.767153 | -0.1744 | $2^{-3}$ | 0.387719 | 0.388474 | 0.1947 |
| $2^{-2}$ | 0.828205 | 0.828492 | 0.0346 | $2^{-2}$ | 0.461290 | 0.462369 | 0.2338 |
| $2^{-1}$ | 0.927273 | 0.928720 | 0.1561 | $2^{-1}$ | 0.577778 | 0.578188 | 0.0710 |
| $2^{-0}$ | 1.070370 | 1.069920 | -0.0421 | $2^{-0}$ | 0.734783 | 0.734253 | -0.0721 |
| $2^{1}$ | 1.240541 | 1.241494 | 0.0769 | $2^{1}$ | 0.906061 | 0.905747 | -0.0346 |
| $2^{2}$ | 1.401754 | 1.403422 | 0.1190 | $2^{2}$ | 1.054717 | 1.054732 | 0.0014 |
| $2^{3}$ | 1.524742 | 1.522220 | -0.1654 | $2^{3}$ | 1.160215 | 1.160321 | 0.0091 |
| $2^{4}$ | 1.603955 | 1.603479 | -0.0297 | $2^{4}$ | 1.224855 | 1.224965 | 0.0089 |
| $2^{5}$ | 1.649555 | 1.649105 | -0.0273 | $2^{5}$ | 1.260961 | 1.258719 | -0.1778 |
| $2^{6}$ | 1.674125 | 1.674843 | 0.0429 | $2^{6}$ | 1.280092 | 1.280384 | 0.0228 |
| $2^{7}$ | 1.686893 | 1.689340 | 0.1451 | $2^{7}$ | 1.289946 | 1.289932 | -0.0011 |
| $2^{8}$ | 1.693403 | 1.693759 | 0.0210 | $2^{8}$ | 1.294948 | 1.294795 | -0.0118 |

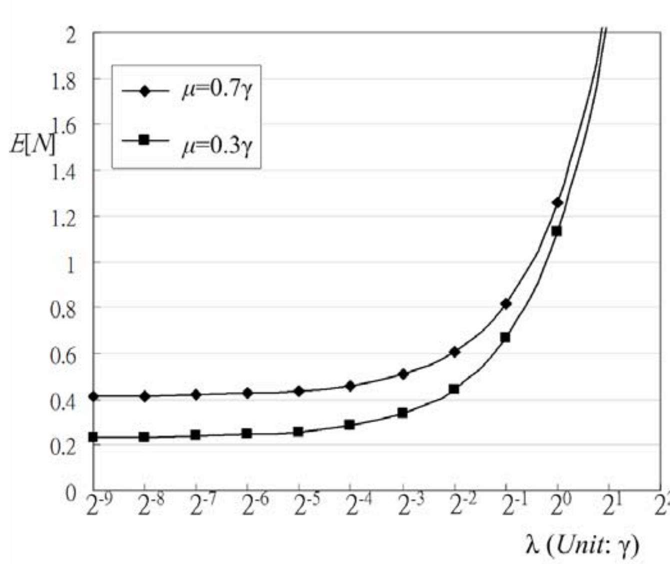
TABLE II
THE $E[N]$ VALUES (ANALYTIC VS. SIMULATION; $\mu$=0.7 $\gamma$ AND 0.3 $\gamma$)

| $\mu$=0.7 $\gamma$ | | | | $\mu$=0.3 $\gamma$ | | | |
|---|---|---|---|---|---|---|---|
| $\lambda/\gamma$ | Analytic | Simulation | Errors (%) | $\lambda/\gamma$ | Analytic | Simulation | Errors (%) |
| $2^{-9}$ | 0.413245 | 0.413348 | 0.0249 | $2^{-9}$ | 0.232376 | 0.231974 | -0.1731 |
| $2^{-8}$ | 0.414727 | 0.414487 | -0.0579 | $2^{-8}$ | 0.233984 | 0.234157 | 0.0739 |
| $2^{-7}$ | 0.417694 | 0.417035 | -0.1577 | $2^{-7}$ | 0.237203 | 0.237027 | -0.0743 |
| $2^{-6}$ | 0.423640 | 0.423137 | -0.1186 | $2^{-6}$ | 0.243654 | 0.242622 | -0.4234 |
| $2^{-5}$ | 0.435582 | 0.435921 | 0.0778 | $2^{-5}$ | 0.256602 | 0.256718 | 0.0452 |
| $2^{-4}$ | 0.459663 | 0.458521 | -0.2485 | $2^{-4}$ | 0.282683 | 0.283195 | 0.1810 |
| $2^{-3}$ | 0.508562 | 0.508762 | 0.0394 | $2^{-3}$ | 0.335526 | 0.335928 | 0.1197 |
| $2^{-2}$ | 0.608974 | 0.608950 | -0.0040 | $2^{-2}$ | 0.443548 | 0.443639 | 0.0204 |
| $2^{-1}$ | 0.818182 | 0.817338 | -0.1031 | $2^{-1}$ | 0.666667 | 0.665715 | -0.1428 |
| $2^{-0}$ | 1.259259 | 1.258305 | -0.0758 | $2^{-0}$ | 1.130435 | 1.130767 | 0.0294 |
| $2^{1}$ | 2.189189 | 2.190114 | 0.0422 | $2^{1}$ | 2.090909 | 2.092014 | 0.0528 |
| $2^{2}$ | 4.122807 | 4.121323 | -0.0360 | $2^{2}$ | 4.056604 | 4.058766 | 0.0533 |

## V. NUMERICAL RESULTS




(a) Effect of $\mu$




(b) Effect of $\lambda$

Fig. 4. Effects of $\mu$ and $\lambda$ on the Expected Number of Lost Calls.

Based on the analytic and simulation models, we use numerical examples to investigate the SPC performance. Based on Equation (6), Fig. 4 plots $E[N]$ against $\mu$ and $\lambda$. It is clear that $E[N]$ increases as the expected *T1* period ($1/\mu$) decreases or the call arrival rate $\lambda$ increases. Consider two extreme cases where $\mu \to 0$ and $\mu \to \infty$. We have

$$\lim_{\mu \to 0} E[N] = \lim_{\mu \to 0} \left[ \frac{(\lambda+\gamma)(\lambda+\mu)}{\gamma(\lambda+\mu+\gamma)} \right] = \frac{\lambda}{\gamma} \quad \text{and}$$

$$\lim_{\mu \to \infty} E[N] = \lim_{\mu \to \infty} \left[ \frac{(\lambda+\gamma)(\lambda+\mu)}{\gamma(\lambda+\mu+\gamma)} \right] = 1 + \frac{\lambda}{\gamma}.$$

$\mu \to 0$ indicates that the length of *T1* approaches infinite, and the first outstanding call keeps on waiting until the SIP UA activation completes. In this case, the outstanding call is always connected, and $E[N]$ approximates to the number of subsequent incoming calls arriving during the SIP UA activation period, which is $\lambda/\gamma$. On the other hand, if $\mu \to \infty$, we have *T1*=0, and *T1* expires immediately after an outstanding call arrives. Therefore, the first outstanding call and all incoming calls arriving during the SIP UA activation are lost. The expected number of such calls is $1+\lambda/\gamma$. These results are clearly observed in Fig. 4 (a).

Fig. 4 (a) also indicates that in order to reduce $E[N]$ in SIP UA activation, it is appropriate to set μ as γ/32. For any μ values smaller than γ/32, the reduction of $E[N]$ is insignificant. Fig. 4 (b) indicates that $E[N]$ is insignificantly affected by the incoming call arrival rate λ when $\lambda<\gamma/32$. This result implies that to ensure good $E[N]$ performance, the SIP UA activation mechanism must be designed such that the activation time is shorter than 0.03125 times of the inter-call arrival time.

## VI. Conclusion

In this paper, we study a WLAN and Cellular network integration solution called WGSN. We proposed an analytic model to investigate the performance of *SIP Push Center* (SPC) [1]. In this mechanism, when a caller sends a SIP request to an MS with its WLAN module being turned off, the SPC sends a GSM short message to activate the SIP UA of the called MS, and a timer *T1* is set for this activation procedure. If *T1* expires before the SIP UA is activated, the call is dropped. If a new incoming call arrives before the outstanding call is complete or timed out, the new incoming call is also dropped. In this paper, we focus on measuring lost calls during SIP UA activation. Our study indicates that *T1* significantly affects the SPC performance. To obtain good $E[N]$ performance, the *T1* period should be longer than 32 times of that for SIP UA activation, and SIP UA activation should be completed within 0.03125 times of the inter-call arrival time.

The SPC utilizes GSM short message service mechanism. According to [5], it takes 20 seconds to transmit a short message to an MS. This transmission delay significantly contributes to the SIP UA activation time and may not be acceptable. Therefore, we may further shorten the SIP UA activation time by using high priority short messages service. Short messages with high priority would be transmitted to the destination within, e.g., 3-5 seconds. This improvement allows SIP UA activation to be complete with an acceptable delay. In the future, we will extend our study to investigate the SPC performance under general SIP UA activation times.

## Appendix. Notation

- $1/\gamma$: the expected SIP UA activation time
- $1/\lambda$: the expected inter incoming call arrival time
- $1/\mu$: the expected *T1* timeout period
- $E[N]$: the expected number of lost calls during SIP UA activation time
- $f_{t_0}(t_0)$: the density function for the $t_0$ distribution
- $f_{t_1}(t_1)$: the density function for the $t_1$ distribution
- $f_{t_2}(t_2)$: the density function for the $t_2$ distribution
- $t_0$ : the inter incoming call arrival time
- $t_1$: the *T1* timeout period
- $t_2$: the SIP UA activation time

## References


[1] W.-S. Feng, L.-Y. Wu, Y.-B. Lin, and W.-E. Chen, "WGSN: WLAN-based GPRS Support Node with Push Mechanism," The Computer Journal, vol. 47, no. 4, pp. 405-417, July 2004.

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

[3] J. Rosenberg et. al., "SIP: Session Initiation Protocol," Technical Report RFC 3261, Internet Engineering Task Force, Jun. 2002.

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

[5] H.-N. Hung, Y.-B. Lin, M.-K. Lu, and N.-F. Peng, "A Statistic Approach for Deriving the Short Message Transmission Delay Distributions," Accepted and to appear in *IEEE Trans. Wireless Communication.*