

# An Effective Failure Recovery Mechanism for SIP/IMS Services

Whai-En Chen, Li-Yao Tseng, Chien-Lung Chu  
Dept. of Computer Science and Information Engineering  
National Ilan University  
I-Lan City, Taiwan, R.O.C  
wechen@niu.edu.tw

**Abstract**—There are many challenges to provide telecommunication services. The reliability is one of the most important challenges. Telecommunications systems require 99.999% reliability, which means that system only tolerant of downtime about 5 minutes per year. To achieve such high reliability, multiple servers and a shared database are deployed for a Session Initiation Protocol (SIP)/ IP Multimedia Subsystem (IMS) service. This paper investigates the effects of different database (DB) access mechanisms such as Write-Through (WT), Write-Back (WB), and the database-only (DB-only) mechanisms. Then this paper proposes an enhancement for the WB mechanism to improve the successful call setup probability. The analytic models are derived to analyze the performance of the proposed mechanism.

**Keywords**- IMS; VoIP; SIP; Write-Back; Write-Through

## I. INTRODUCTION

Voice communication has already been a necessary service for a long time in human's daily life. Due to the rapid development and deployment of the broadband Internet Protocol (IP) networks (i.e., Internet), voice can also be transmitted through IP networks referred to as Voice over IP (VoIP) [1]. Most of the VoIP systems adopt the *Session Initiation Protocol* (SIP) [2] as the signaling protocol to establish the upcoming multimedia streams, and the media such as voice and video are transmitted by *Real-time Transport Protocol* (RTP). However, the reliability of VoIP is one of the most important challenges. Specifically, the telecommunications systems require 99.999% reliability, which means that system only tolerant of downtime about 5 minutes per year [3]. There are many failure types such as hardware failure, operation system failure, network failure and service failure. This paper focuses on handling the service failure. The network failure is not in the scope of this paper and the readers are referred to [4][5] for further study. To achieve the reliability, many research [6][7][8][9][10] propose various designs to achieve high availability and reliability for SIP-based VoIP services. The backup mechanisms, which stored the records in a shared database, are a widely used to enhance the reliability. Typically, two or more SIP servers are deployed in the backup mechanisms. One of the SIP servers is the master server, which stores the records into the shared

database. After the master SIP server's failure occurs, the slave SIP servers are notified. One of the slave SIP servers becomes a new master SIP server, retrieves the UAs' records from the database, and then takes over the VoIP service. In this paper, we investigate the *Write-Through* (WT), *Write-Back* (WB), and the *database-only* (DB-only) mechanisms. Among these mechanisms, the WB mechanism has the lowest registration latency and costs the lowest forwarding latency. However, the registration records in the WB mechanism may be lost if the records are not written into the DB before the server's failure. In this paper, we propose a *WB with Lost Registration Recovery* (WB-LRR) mechanism to enhance the successful call setup probability.

In the following sections, the WT, WB, and DB-only mechanisms are described in section 2. The analysis of these mechanisms and an improved WB mechanism are elaborated in section 3. The analytic models are conducted and then the performance is analyzed in section 4. Finally, the conclusions are given in section 5.

## II. THE SIP SERVER'S OPERATION FOR HIGH AVAILABILITY

If the VoIP system is served by only one SIP server, the service may be interrupted when the server is failed. To solve the problem, multiple SIP servers are typically deployed for backup. When the master SIP server failed, one of the slave SIP servers becomes the new master SIP server and takes over the VoIP service. The new master SIP server needs to retrieve the UAs' registration records before the arrival of the next incoming call to prevent the call setup failure.

Typically, the SIP server stores the UAs' registration records in both the local memory of the master SIP server and/or the shared DB. The Write-Through (WT), Write Back (WB) and DB-only mechanisms are widely used to ensure the UAs' records are successfully stored into the DB and achieve high reliability.

To analyze these mechanisms, we illustrate the registration procedures and the call setup procedures for the WT, WB and DB-only, respectively. To focus on the mechanisms, this paper illustrates the procedures without the SIP authentication process.

---

The work of Whai-En Chen was sponsored in part by MOST 103-2221-E-197 -004, 103-2218-E-009 -029 projects and III project.

### A. Registration procedures

- The Write-Through mechanism

At Step 1, the UA issues a SIP REGISTER message to the SIP server.

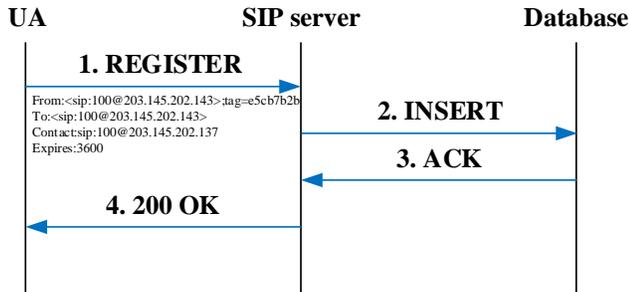


Figure 1. SIP registration message flow in the Write-Through mechanism

The REGISTER message contains the registration information in the SIP header fields, including the *From*, *To*, *Call-ID*, *Contact*, and *Expires* header fields. *From* header field indicates the address-of-record of the UAC (registering user). *To* header field indicates the address-of-record of the UAS. *Call-ID* header field is a unique identifier for the dialog session. *Contact* header field provides a *Uniform Resource Identifier* (URI) for further communications. *Expires* header field records the lifetime of the address-of-record [2]. The SIP server retrieves the UA’s information (e.g., address-of-record and the IP/port information) from the SIP REGISTER message and then stores the information into its local memory (i.e., RAM) and the DB. In order to identify the synchronization status of the records, the *State* field is used to maintain the status of the UA’s registration information [11]. The value of the *State* field can be *New*, *Synced* or *Dirty*. The *New* state means that the UA’s registration record is only stored in the local memory but not stored in the DB yet. The *Synced* state means that the UA’s registration record is written to the DB. In this state, the contents in the memory and the DB are the same. The *Dirty* state means that UA’s registration information in the memory differs from that in the DB.

At Step 2, according to the *State* field, the SIP server can utilize the correct SQL command such as **INSERT** to add record when the state is *NEW* or **UPDATE** to refresh record when the state is *Dirty* in the DB. Assume that SQL **INSERT** command is used. At Step 3, if the command in the Step 2 is executed successfully, then the DB sends an **ACK** message to the SIP server. At Step 4, when the SIP server receives the **ACK** message, it replies a SIP **200 OK** message to the UA to indicate the registration procedures is successful. Note that the SIP UA sends the **REGISTER** message to the SIP server periodically or when the SIP UA obtains a new IP address.

- The Write-Back mechanism

The registration procedure of the WB mechanism is similar to that of the WT mechanism. The UA sends a SIP REGISTER message to the SIP server at Step 1. When the SIP server stores the UA’s information in its local memory, it

immediately replies a SIP **200 OK** message to indicate the registration procedures is successful at the Step 2.

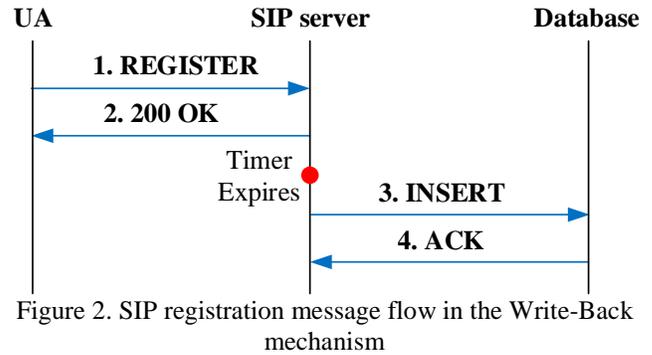


Figure 2. SIP registration message flow in the Write-Back mechanism

At Step 3, the SIP server does not immediately store the records into the DB until the timer expires. In this example, the SIP server retrieves the records from its local memory and stores them into the DB by the SQL **INSERT** command. At Step 4, the DB replies an **ACK** message to indicate that the UAs’ registration records have been written to the DB. Then the SIP server changes the states of the UAs’ records in the local memory to the *Synced* state.

- The DB-only mechanism

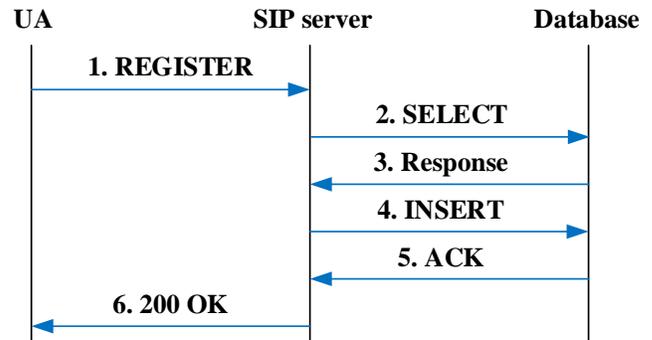


Figure 3. SIP registration message flow in DB-Only mechanism

The UA sends a SIP REGISTER message to the SIP server at Step 1. In Steps 2-5, the SIP server stores the UA’s information into the DB. In the DB-only mechanism, all the UA’s records are stored in the DB and there is no records in the local memory. When the SIP server receives the REGISTER message, the SIP server queries the DB whether the records have been stored in the DB. If yes, the SIP server utilizes the SQL **UPDATE** command to refresh the records. Otherwise, the SIP server utilizes the SQL **INSERT** command to add the records.

At Step 2, the SIP server adopts the SQL **SELECT** command to search whether the target is in the DB or not. At Step 3, the DB replies the SIP server that the records do not exist. At Step 4, the SIP server uses the SQL **INSERT** command to add the UA’s registration information into the DB. At Step 5, the DB replies an **ACK** message to indicate that the UA’s registration information has successfully stored into the DB. At Step 6, the SIP server replies a **200 OK** message to indicate the registration procedures is successfully completed.

B. Call setup procedures

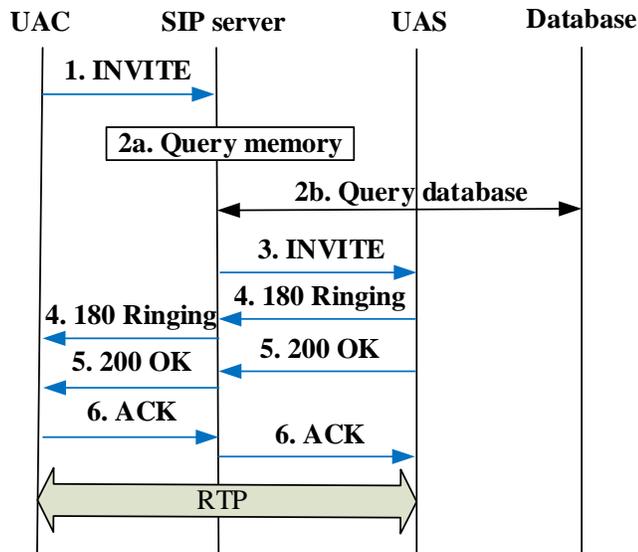


Figure 4. The call setup message flow

When the SIP server receives the **INVITE** message, it checks the registration records from either the local memory in the WT/WB mechanism or the DB in the DB-only mechanism to find the IP address and port of the User Agent Server (UAS, the called party). The call setup procedures is illustrated in Figure 4 and described in detail as follows:

- Step 1.** The User Agent Client (UAC, the calling party) sends a SIP **INVITE** message to the UAS through the SIP server. The SIP server checks the access mechanism (WT, WB or DB-only) of the DB. If the SIP server adopts the WT or WB mechanism, the SIP server implements **Step 2a**. Otherwise (i.e., DB-only), the SIP server implements **Step 2b**.
- Step 2a.** The SIP server retrieves the UAS’s information (e.g., IP address and port) from its main memory.
- Step 2b.** The SIP server queries the UAS’s information (e.g., IP address and port) from the DB. The DB replies the results to the SIP server through a response message.
- Step 3.** Assume that the UAS’s information is obtained successfully at **Step 2a** or **Step 2b**, the SIP server refreshes the *Request-URI* field in the **INVITE** message. Then, according to the result from **Step 2a** or **Step 2b**, the SIP server forwards the **INVITE** message to the UAS. Otherwise, the SIP server replies a *404 Not Found* message to the UAC.
- Step 4.** When the UAS receives the **INVITE** message, it plays the ringing tone to notify the called party and replies a **180 ringing** message back to the UAC through the SIP server. Upon receipt of the **180 ringing**, the UAC plays

the ring-back tone.

**Steps 5-6.** The UAS sends a **200 OK** message to the UAC when the called party accepts the call. Upon receipt of the **200 OK** message at **Step 5**, the UAC replies an **ACK** message back to the UAS. At this point, the call setup procedures are completed. The UAC and UAS can communicate with each other through RTP.

III. THE ENHANCED WRITE-BACK MECHANISM

A. Failover procedures and Comparison of the mechanisms

In the WT/WB mechanisms, the SIP server stores the UAs’ information in both its local memory and the shared DB. When the monitoring system detects the failure occurs, it automatically notifies the slave server and switches the traffic to the slave servers. Upon receipt of the notification from the monitoring system, the slave SIP server retrieves the UAs’ registration information from the DB and stores in its main memory. Then, the slave SIP server changes its role to be the master SIP server and takes over the VoIP service.

The SIP server stores all UAs’ registration records in the DB when the DB-only mechanism is adopted. Therefore, after the slave SIP server becomes the master SIP server, it takes time to re-establish the connections to the DB. If the slave SIP servers establish the connections to the DB before the master SIP server is failure, the registration latency can be reduced.

The registration latency contains the SIP processing latency, the DB access latency and the transmission latency. Either in the WT or the DB-only mechanism, the SIP server stores the registration record into the DB and then replies the **200 OK** message. The SIP server can select the Write-Back mechanism to reduce the registration latency. When the SIP server receives a **REGISTER** message in the WB mechanism, the SIP server stores the registration information in its main memory and immediately replies a **200 OK** message. After a timer expired, the SIP server stores the buffered registration information to the DB. However, in the WB mechanism, before the timer is expired, the UAs’ registration records in the local memory are not updated into the DB. If the server is failed in this case, the registration records in the local memory are lost.

In Table I, we compare the WT, WB and DB-only mechanisms in terms of the update method, the storage of UAs’ records, registration latency and forwarding latency. The SIP server requires the UA to update its state periodically in the WB mechanisms. The DB-only mechanism does not buffer the UAs’ registration information in the main memory.

We also compare the different in the response latency. The response latency can be the registration latency or the forwarding latency. Then, the registration latency is that the UA sends a **REGISTER** message until receives the

Table I. Comparison of the WT, WB and DB-only mechanisms

Mechanism	Update Method	Storage of UAs’ records	Registration Latency	Forwarding Latency
WT	Immediate Update	Main Memory and DB	Long	Short
WB	Periodic Update	Main Memory and DB	Short	Short
DB-only	Immediate Update	DB	Longest	Long

corresponding **200 OK** message. The forwarding latency is that from the SIP server receives the **INVITE** message (sending by the calling party, UAC) to the SIP server forwards the **INVITE** message.

The SIP server utilizes the WB mechanism can reduce the response latency but may increase the lost rate. Thus, we propose a method to improve the WB mechanism to reduce the probability that the registered records are lost. By improving the WB mechanism, the SIP server can reduce the lost rate of the registration records while keeping the shorter response latency.

### B. The proposed Write-Back Mechanism

In the Write-Back (WB) mechanism, the UAs' registration information is stored in the SIP server's main memory (i.e., RAM) and is not updated into the DB before the timer expires. In this mechanism, the SIP server can quickly reply a **200 OK** message. Therefore, the response time of the WB mechanism is faster than the WT and DB-only mechanisms. However, when the master server is failed, the new master SIP server can't retrieve the UAs' registration records from the DB if the registration records are not stored in the DB. Thus, the WB mechanism is less dependable than the WT and DB-only mechanisms. The heartbeat mechanism [6][7] is used for detecting the server's failure. If the slave SIP servers do not receive the heartbeat message from the master SIP server for a while, the slave servers assume the master server is failed. The slave SIP server with the highest priority changes its role to be the new master SIP server and takes over service. Then, the new master SIP server tries to restore the registration UAs' records from the DB. In the WB mechanism, the UAs' registration records with the state *New* or *Dirty* are not be written into the DB. If the master SIP server is failed in this case, the records cannot be retrieved by the new master SIP server. In other words, some of the registered UAs cannot be reached before the UAs register to the new master SIP server.

To improve the reliability of the WB mechanism, we propose a WB with Lost Registration Recovery (WB-LRR) mechanism to recover the lost UAs' registration records. In the WB-LRR mechanism, we propose the SIP server not only buffers the UAs' records in the main memory but also in its hard disk. When the timer expires, the master SIP server stores the UAs' records with the *New* or *Dirty* state into the DB. If the records are stored successfully, the SIP server changes the records' state to *Synced* and erases the records from the hard disk.

When the master SIP server is failed, the slave SIP server becomes the new master server, takes over the VoIP service and notifies the slave SIP server. After the old master SIP server restarts, it changes its role to be a slave server and checks its hard disk. If the UAs' records exist in the hard disk of the old master SIP server, the server acquires the IP address and port of the new master SIP server from the DB. Then the old master SIP server sends the UAs' records to the new master SIP server by sending the **REGISTER** messages. In the WB-LRR, the old master SIP server acts a registration agent for the UAs. In the **REGISTER** message, the *from* header records the old SIP server's address-of-record. The expired time in the *Expires* header is the original time minus the

SIP server's restarted time. *To*, *Call-ID* and *Contact* headers are not modified. When the new master server receives the **REGISTER** message from the restarted server, it checks whether the expired time of the address-of-record is greater than the records stored in its local memory. If yes, the master server updates the UAs' registration records. If not, the master server ignores the UAs' registration. After that the new master SIP server replies the **200 OK** messages to the restarted SIP server to complete the WB-LRR process.

## IV. ANALYTIC MODELS AND NUMERIC RESULTS

To obtain the quantity sections, we derive the analytic models for evaluating the analyze reliability of the VoIP system. Specifically, we evaluate the probability that the UA's registration is written into the DB in the periodic update and the probability of the successful incoming call after the SIP server failure. The input parameters used in this section are listed in Table II.

Table II. Notation (Input Parameters)

$\mu$	the periodic update frequency
$\lambda$	the registration arrival rate
$\gamma$	the incoming call arrival rate
$V$	the variance of the inter-registration
$1/\delta$	the time when the master SIP server receives the WB-LRR

### A. The successful call after the SIP server's failure

In this section, we analyze the UA's registration, the server's periodic update, the incoming call activities and the server's failure event to obtain the probability that the calling party (i.e., UAC) successfully makes a call to the called party (i.e., UAS) after the SIP server's failure occurs. Let  $p_{WB}$  and  $p_{WB-LRR}$  be the probability that the UAC successfully makes a call to the UAS after the SIP server's failure occurs by using the WB (Write-Back) and the WB-LRR (WB-Lost Registration Recovery) mechanism respectively. In this subsection, we derive the probabilities ( $p_{WB}$  and  $p_{WB-LRR}$ ) that the SIP server successfully sets up the call. The UA's registration should be updated to the DB (**case 1**) or the UA re-registers to the new SIP server before the incoming call arrives (**case 2**) when utilizing the WB mechanism. By using the proposed WB-LRR mechanism, the UA's registration is transferred to the new master SIP server by the recovered SIP server (**case 3**). Thus, similar to the WB mechanism, the call can be successfully established in **case 1**, **case 2** or **case 3**.

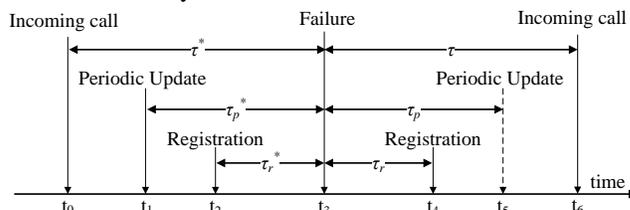


Figure 5. Timing diagram before and after a SIP server's Failure

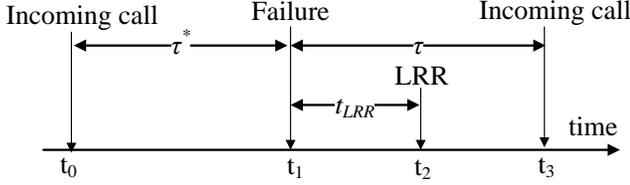


Figure 6. Timing diagram for WB-LRR and incoming call

Consider the timing diagram in Figure 5. The incoming calls arrive at  $t_0$  and  $t_6$ , and the registrations arrive at  $t_2$  and  $t_4$ . The periodic update is performed at  $t_1$  and will be performed at  $t_5$ . Since the registration ( $t_2$ ) arrives after the periodic update ( $t_1$ ), the registration is temporarily stored in the local memory and should be stored into the shared DB in the next periodic update ( $t_5$ ). On the other hand, if the SIP UA registers to the SIP server ( $t_4$ ) before the incoming call arrives at the SIP server ( $t_6$ ), the SIP server can find the records of the UA no matter the records are written into the DB or not.

Assume that the failure occurs in the SIP server at  $t_3$ . Thus, the registration at  $t_2$  is not updated to the DB. If the UA re-registers at  $t_4$ , the incoming call at  $t_6$  can be processed successfully. Otherwise, the SIP server cannot find the records of the UA or obtain the incorrect records, and the call cannot be established successfully.

Then, we assume  $p_b$  to represent the probability that the UA's registration records are lost after the SIP server's failure in **Case 1**. Assume that the failure occurs in the SIP server at  $t_3$ . The last periodic update before the SIP server's failure occurs at  $t_1$ . The last registration before the SIP server's failure occurs at  $t_2$ . A SIP server's failure  $t_3$  is a random observer of the periodic update interval times and the inter-registration times.  $\tau_p^*$  is the periodic update interval time.  $\tau_r^*$  is the inter-registration time. Based on the excess-life theorem,  $\tau_p^* = t_3 - t_1$  has an exponential distribution with mean  $1/\mu$ , and  $\tau_r^* = t_3 - t_2$  has a distribution function  $R(\cdot)$ , density function  $r(\cdot)$ , and Laplace transform  $r^*(s)$ , based on the excess-life theorem [12], we obtain

$$r(\tau_r^*) = \lambda [1 - F(\tau_r^*)]$$

and

$$\begin{aligned} r^*(s) &= \int_{\tau_r^*=0}^{\infty} r(\tau_r^*) e^{-s\tau_r^*} d\tau_r^* \\ &= \left( \frac{\lambda}{s} \right) [1 - f^*(s)] \end{aligned} \quad (1)$$

Based on (1), we derive  $p_b$  as

$$\begin{aligned} p_b &= \Pr[\tau_p^* > \tau_r^*] \\ &= \int_{\tau_p^*=0}^{\infty} \int_{\tau_r^*=0}^{\tau_p^*} r(\tau_r^*) \mu e^{-\mu\tau_p^*} d\tau_r^* d\tau_p^* \\ &= \int_{\tau_p^*=0}^{\infty} R(\tau_p^*) \mu e^{-\mu\tau_p^*} d\tau_p^* \end{aligned}$$

$$\begin{aligned} &= \frac{\mu r^*(s)}{s} \Big|_{s=\mu} \\ &= \left( \frac{\lambda}{\mu} \right) [1 - f^*(\mu)] \end{aligned} \quad (2)$$

If the UA's registration comes before the incoming call after the failure, the UA can be reached no matter the registration is lost or not. Let  $p_c$  be the probability that a registration comes before an incoming call after a SIP server's failure occurs. In Figure 5, Let  $\tau = t_6 - t_3$  and  $\tau_r = t_4 - t_3$ . A SIP server's failure  $t_3$  is a random observer of the inter-call arrival times and the inter-registration times, and  $\tau$  is the excess life of an inter-call-arrival time.  $\tau_r$  is the excess life of an inter-registration time. Assume that the call arrivals to the UA follow a Poisson process with the rate  $\gamma$ . Based on the excess-life theorem,  $\tau$  has an exponential distribution with the mean  $1/\gamma$ . Similarly,  $\tau_r$  has a distribution function  $R(\cdot)$ , density function  $r(\cdot)$ , and Laplace transform  $r^*(s)$ . Similar to the derivation for (2)

$$\begin{aligned} p_c &= \Pr[\tau > \tau_r] \\ &= \left( \frac{\lambda}{\gamma} \right) [1 - f^*(\gamma)] \end{aligned} \quad (3)$$

We analyze the  $p_d$  in **Case 3**. In Figure 6, we consider that the proposed WB-LRR occurs before the incoming call. Assume that a SIP server's failure occurs at  $t_1$ . The last incoming call occurs at  $t_0$ . The WB-LRR after the failure occurs at  $t_2$ . The first incoming call after the failure occurs at  $t_3$ .  $\tau$  is the excess life of an inter-call-arrival time. The  $T_{LRR}$  is a random variable with mean  $1/\delta$ . Assume that the call arrivals to the UA are a Poisson process with the rate  $\gamma$ . Thus,  $\tau$  has an exponential distribution with mean  $1/\gamma$ .

$$\begin{aligned} p_d &= \Pr[\tau > T_{LRR}] \\ \Pr[X = x, T = t] &= \left[ \frac{(\gamma t)^x}{x!} \right] e^{-\gamma t} \\ &\text{and} \\ p_d &= \Pr[\tau > T_{LRR}] \\ &= \Pr\left[X = 0, T = \frac{1}{\delta}\right] \\ &= e^{-\frac{\gamma}{\delta}} \end{aligned} \quad (4)$$

The probability  $p_{WB}$  is derived as follows. From (2) and (3), the probability  $p_{WB}$  is expressed as

$$\begin{aligned} p_{WB} &= 1 - p_b(1 - p_c) \\ &= 1 - \left\{ \left( \frac{\lambda}{\mu} \right) [1 - f^*(\mu)] \right\} \left\{ 1 - \left( \frac{\lambda}{\gamma} \right) [1 - f^*(\gamma)] \right\} \end{aligned} \quad (6)$$

Assume that the Gamma distribution with mean  $1/\lambda$ , variance  $V$ , and Laplace transform as expressed in (7).

$$f^*(s) = \left( \frac{1}{V\lambda s + 1} \right)^{\frac{1}{V\lambda^2}} \quad (7)$$

From (7), we have

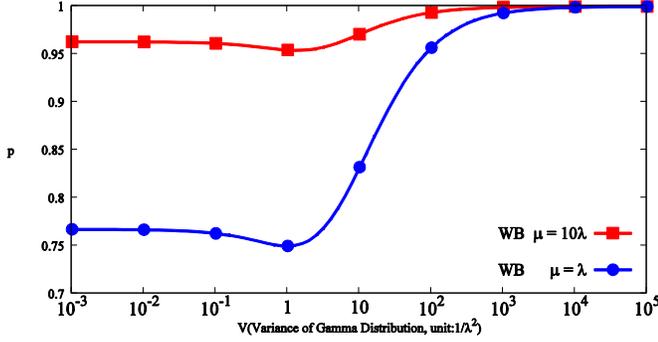
$$1 - \left( \frac{\lambda^2}{\mu\gamma} \right) \left[ 1 - \left( \frac{1}{V\lambda\mu + 1} \right)^{\frac{1}{V\lambda^2}} \right] \left[ \frac{\gamma}{\lambda} - 1 + \left( \frac{1}{V\lambda\gamma + 1} \right)^{\frac{1}{V\lambda^2}} \right] \quad (8)$$

We obtain the probability  $p_{WB-LRR}$  from **Case 1**, **Case 2** and **Case 3**. The probability  $p_{WB-LRR}$  is derived as follows. From (2), (3), and (5), the probability  $p_{WB-LRR}$  is expressed as

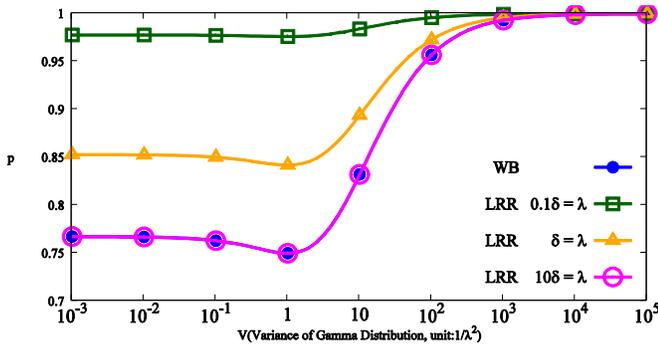
$$p_{WB-LRR} = 1 - p_b(1 - p_c)(1 - p_d) = 1 - \left\{ \left( \frac{\lambda}{\mu} \right) [1 - f^*(\mu)] \right\} \left\{ 1 - \left( \frac{\lambda}{\gamma} \right) [1 - f^*(\gamma)] \right\} \left( 1 - e^{-\frac{\gamma}{\delta}} \right) \quad (9)$$

If  $f^*(s)$  is a Gamma Laplace transform as expressed in (7), then from (9), we have

$$p_{WB-LRR} = 1 - \left( \frac{\lambda^2}{\mu\gamma} \right) \left[ 1 - \left( \frac{1}{V\lambda\mu + 1} \right)^{\frac{1}{V\lambda^2}} \right] \left[ \frac{\gamma}{\lambda} - 1 + \left( \frac{1}{V\lambda\gamma + 1} \right)^{\frac{1}{V\lambda^2}} \right] \left( 1 - e^{-\frac{\gamma}{\delta}} \right) \quad (10)$$



(a) Comparison of WB ( $r = \lambda$ )



(b) Comparison of WB and WB-LRR ( $r = \mu = \lambda$ )

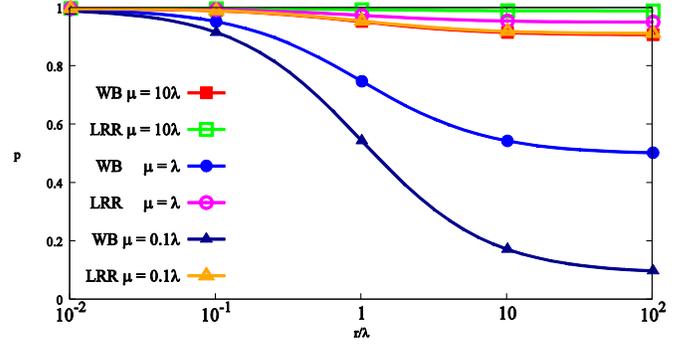
Figure 7. Effects of the variance of the inter-registration intervals

Figure 7 shows the probabilities  $p_{WB}$  and  $p_{WB-LRR}$  for the Gamma-distributed inter-registration intervals with different variance values. In Figure 7, the  $p_{WB}$  increase when  $\mu$  increase. The probabilities  $p_{WB}$  and the  $p_{WB-LRR}$  are very close when  $\delta$  is ten times slower than  $\gamma$ . When the small variance  $V$ , the  $p_{WB-LRR}$  is significantly increased.

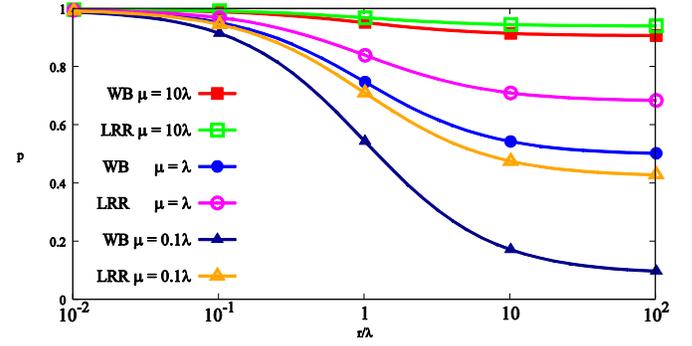
When the interval  $t_r$  is exponentially distributed (i.e.,  $V = 1/\lambda^2$ ), the probabilities  $p_{WB}$  and  $p_{WB-LRR}$  of the formula are derived as follows

$$p_{WB} = 1 - \left( \frac{\lambda}{\mu + \lambda} \right) \left( \frac{\gamma}{\gamma + \lambda} \right) \quad (11)$$

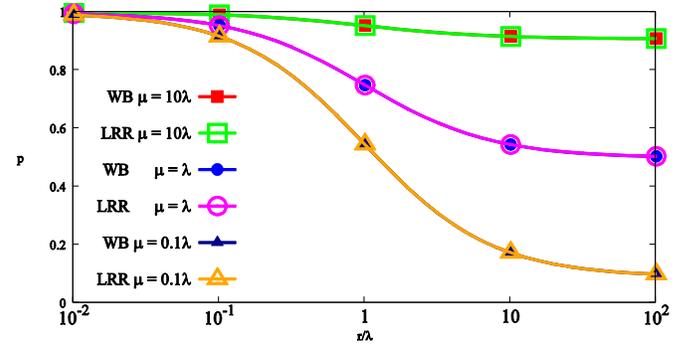
$$p_{WB-LRR} = 1 - \left( \frac{\lambda}{\mu + \lambda} \right) \left( \frac{\gamma}{\gamma + \lambda} \right) \left( 1 - e^{-\frac{\gamma}{\delta}} \right) \quad (12)$$



(a).  $\gamma / \delta = 0.1$



(b).  $\gamma / \delta = 1$



(c).  $\gamma / \delta = 10$

Figure 8. Successfully sets up the call after a SIP server's failure ( $V = 1/\lambda^2$ )

We observe the effects on the different rates of the WB-LRR mechanism in Figure 8(a), (b) and (c). Figure 8(a) draws  $p_{WB}$  and  $p_{WB-LRR}$  when  $\delta$  is ten times faster than  $\gamma$ . In this case, the  $p_{WB}$  of the worst case ( $\mu = 0.1\lambda$ ) is limited to 9%. The  $p_{WB-LRR}$  of the worst case is limited to 91%. The

## REFERENCES

probability significantly increases 82%. In Figure 8(b),  $\delta$  is the same as  $\gamma$ . The  $p_{WB-LRR}$  of the worst case is limited to 42%. The probability increase 33%.  $\delta$  is 10 times slower than  $\gamma$  in Figure 8(c). The probabilities  $p_{WB}$  and the  $p_{WB-LRR}$  are very close.

To increase the call success rate, the restarted slave SIP server should send the **REGISTER** messages with the buffered records to the master SIP server as soon as possible in the proposed WB-LRR mechanism. Then the probability of the call success rate is significantly increased.

## V. CONCLUSIONS

To reduce the response time, the Write-Back (WB) mechanism is selected as the backup mechanism for SIP/IMS services. However, in the WB mechanism, the SIP server does not store the UAs' records until the timer expired. Thus, the UAs' registration records buffered in the main memory may be lost when the SIP server failure occurs. We propose the WB with Lost Registration Recovery (WB-LRR) mechanism to enhance the probability of the successful calls. Through the numerical analysis, the results show that the probability of the call success rate is significantly increased. That's because the restarted SIP server in WB-LRR stores the buffered registrations to the new master SIP server and the shared database by sending the REGISTER messages. The WB-LRR mechanism provides quick response latency for registration and call setup procedures and offers higher successful calling probability.

- [1] "VoIP," [http://en.wikipedia.org/wiki/Voice\\_over\\_IP](http://en.wikipedia.org/wiki/Voice_over_IP)
- [2] J. Rosenberg, H. Schulzrinne, Columbia U, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, IETF, Jun 2002.
- [3] "High availability," [http://en.wikipedia.org/wiki/High\\_availability](http://en.wikipedia.org/wiki/High_availability)
- [4] J. Barakovic, H. Bajric, M. Kos, "Priority Transmission of SIP Signaling Flows in Case of Link Failure," Proceedings of the 10th International Conference on Telecommunications ConTEL 2009, pp. 389-396, Jun 2009.
- [5] S. N. Avci, Xiaodan Hu, E. Ayanoglu, "Recovery from Link Failures in Networks with Arbitrary Topology via Diversity Coding," 2011 IEEE Global Telecommunications Conference – GLOBECOM 2011, pp. 1-6, Dec 2011.
- [6] Wei-Ming Wu, Kuochen Wang, Rong-Hong Jan, and Chia-Yuan Huang, "A Fast Failure Detection and Failover Scheme for SIP High Availability Networks," 13th Pacific Rim International Symposium on Dependable Computing PRDC 2007, pp. 187-190, Dec 2007.
- [7] Yun-Jung Cheng, Kuochen Wang, Rong-Hong Jan, Chien Chen, and Chia-Yuan Huang, "Efficient failover and Load Balancing for dependable SIP proxy servers," IEEE Symposium on Computers and Communications 2008 ISCC 2008, pp. 1153-1158, Jul 2008.
- [8] I. Kuzminykh, "Failover and load sharing in SIP-based IP telephony," Modern Problems of Radio Engineering, Telecommunications and Computer Science Proceedings of International Conference TCSET 2008, pp. 420-422, Feb 2008.
- [9] Qingwei Yang, Zu-Kuan Wei, Jae-Hong Kim, "A hot backup sip server system based on VRRP," 6th International Conference on Networked Computing and Advanced Information Management NCM 2010, pp. 17-20, Aug 2010.
- [10] Cisco Inc., "High-Availability Solutions for SIP Enabled Voice-over-IP Networks," [http://www.cisco.com/en/US/tech/tk652/tk701/technologies\\_white\\_paper\\_r09186a00800a9818.shtml](http://www.cisco.com/en/US/tech/tk652/tk701/technologies_white_paper_r09186a00800a9818.shtml)
- [11] N. Ohlmeier, "Design and implementation of a high availability SIP server architecture," Thesis, Computer Science Department, Technical University of Berlin, Jul 2003.
- [12] Sheldon M. Ross, Introduction to Probability Models Tenth Edition, Dec 2009.