Implementation and Performance Evaluation for Mobility Management of a Wireless PBX Network

Phone Lin and Yi-Bing Lin, *Senior Member, IEEE*

***Abstract*—As wireless technology advances, wireless products are integrated with enterprise networking to offer cordless terminal mobility. Most corporations have deployed wireless PBXs at the departmental level. However, the mobility management mechanism that integrates these facilities at the corporation level may not be available. This paper describes a mobility management mechanism for an enterprise wireless telephone network. We show how to modify the call model of the private branch exchange (PBX) to accommodate mobility management for an enterprise network. Our design was implemented on a commercial PBX product called Jupiter. An analytical model is proposed to evaluate the performance of the implemented system. Our study shows that with a large number of WPBXs and long Internet message delays, the misrouting probability can be limited to within 1%. This performance result is considered satisfactory.**

***Index Terms*—Handset call routing, mobility management, registration, wireless PBX.**

I. Introduction

MANY corporate employees are away from their assigned wired phones but are still in their offices, either at other locations in the same building or in other buildings of the corporation. As wireless technology advances, wireless products are integrated with enterprise networking to provide employee mobility (the so-called cordless terminal mobility) in the company. Unlike its public cellular counterpart, the enterprise wireless telephone network [1] is free from the air time charges, which provides employees the access to the mobile telephone service within the corporation. An enterprise wireless telephone network can be divided into two levels: the *departmental level* and the *corporation level*.

Departmental Level: In a building, the wireless or wire-line voice/data traffic is exchanged within the same PBX. The PBX is deployed to connect the telephone links from the office building to the public switched telephone network (PSTN). If calls occur between the internal lines, then call routing is handled by the PBX without involving the PSTN. A wireless PBX (WPBX) connects radio base stations to provide mobility at the departmental level. (We assume that the reader is familiar with WPBX. Details of WPBX can be found in [3].)

Corporation Level: For a large corporation with multiple locations, the private lines or virtual networks are used to provide hard-wired connections among corporate locations. These leased lines are billed to the company on a flat, monthly basis. To offer mobility at the corporation level, mobility management is required to coordinate the WPBXs connected by the enterprise network.

Many corporations have deployed WPBXs at the departmental level. However, integration of these facilities at the corporation level may not be available. In this paper, we describe a mobility management mechanism for an enterprise wireless telephone network. This mechanism utilizes the Internet to deliver mobility management messages among WPBXs, which can be implemented with minor modifications to the call model of the WPBXs. Compared to the mobility management in public cellular networks (e.g., IS-41 [4] and GSM MAP [5] that utilize complex SS7 protocols), our mechanism is simpler and easier to deploy. Our design has been implemented on a commercial PBX product called Jupiter (a trademark of Wincomm Corp.).

Fig. 1 shows our mobility management architecture with three WPBXs. The WPBXs are connected to PSTN through digital trunks [see (1) in Fig. 1]. The control or signaling messages for mobility management are delivered through the Internet or Intranet [see (2) in Fig. 1]. A mobile handset accesses the telephone services through the base station (BS) connected to the WPBX [see (3) in Fig. 1]. The mobility management procedure works as follows.

When a handset p moves from WPBX A to WPBX B, it makes a registration request to WPBX B. Then WPBX B announces the location of p to other WPBXs by broadcasting the location update messages. When a call termination for p arrives at a WPBX, the WPBX routes the call to the correct WPBX (where p resides) according to the routing information. Details for mobility management will be described in Sections III and IV. Our current implementation consists of two WPBXs, which can be easily extended to include more WPBXs.

In this paper, we first describe the basic call switching model (BCSM) for Jupiter. Then we show how to modify the call model to accommodate mobility management. An analytical model is proposed to evaluate the performance of the implemented system.

II. Basic Call Switching Model

The Jupiter *basic call switching model* (BCSM) is based on ITU-T Q.931 specification [7], and the PBX software was implemented in C++. Consider a call from subscriber A (the calling party) to subscriber B (the called party). Fig. 2 illustrates the BCSM state diagram for the calling party (A party). Several states are defined in the state diagram.

`QNull` means that no call exists.

Manuscript received June 1, 1999; revised May 1, 1999 and January 16, 2001.
The authors are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: liny@csie.nctu.edu.tw).
Publisher Item Identifier S 0733-8716(01)04206-8.

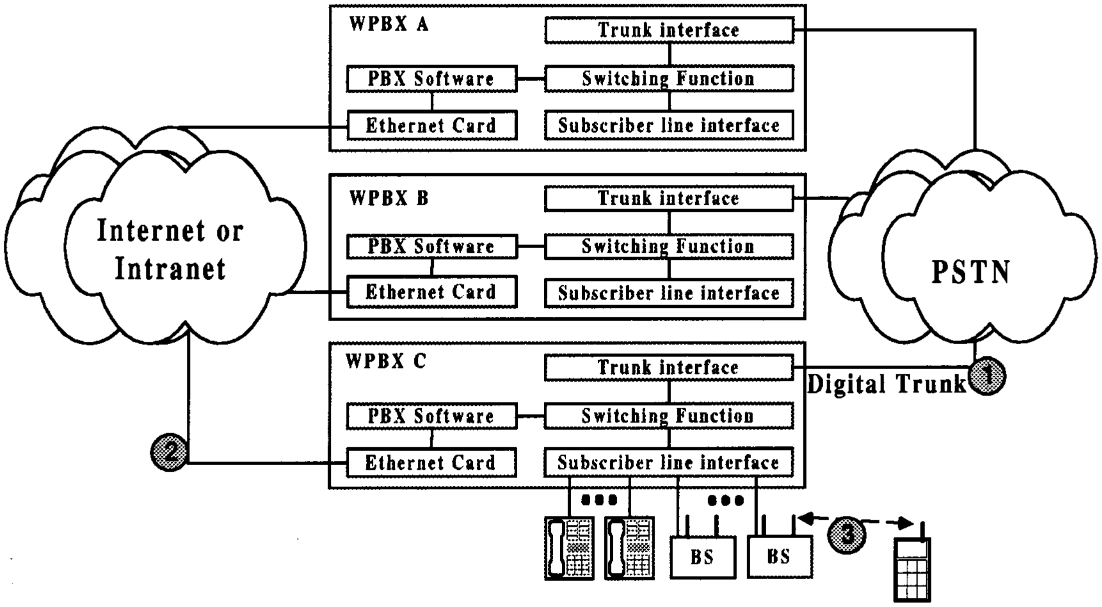

Fig. 1. Mobility management architecture with three WPBXs.

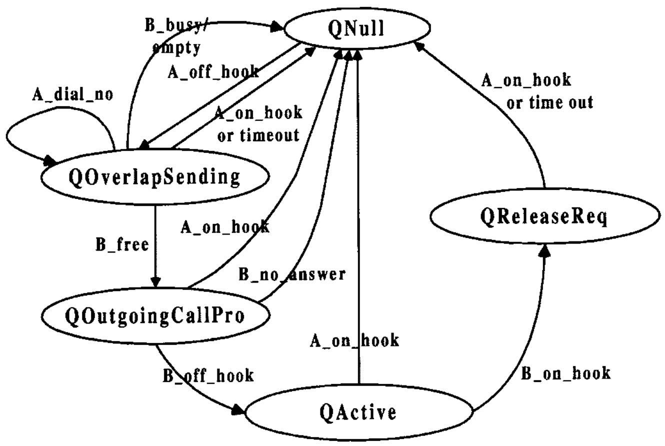

Fig. 2. A simplified BCSM state diagram for calling party (A denotes calling party; B denotes called party).

`QOverlapSending` means that the PBX has acknowledged the call establishment request and has prepared to receive additional call information (if any) from the calling party.

`QOutgoingCallPro` means that the PBX has sent an acknowledgment to the calling party to indicate that all information necessary to effect call establishment has been received.

`QActive` means that the PBX has awarded the call to the called party, and the call path between the calling and called parties has been established.

`QReleaseReq` means that the PBX has requested the corresponding user (either the calling party or the called party) to release the call and is waiting for a response.

Based on the state diagram in Fig. 2, call setup for the calling party is described in the following steps.

Step I—Calling Party Off-Hook: Initially the calling party BCSM is at state `QNull`; i.e., no call presents. Suppose that the calling party picks up the phone. The PBX detects the off-hook situation, creates a call record for the calling party, and sends a dial tone to the calling party. In the call record, the calling party's BCSM state *Astate* is set to `QOverlapSending` to reflect the transition in BCSM.

Step II—Telephone Number Dialing: After receiving the dial tone, the calling party starts dialing the telephone number of the called party. Since the calling party is at state `QOverlapSending`, the `QOverlapSending` function in Fig. 3 is invoked when a dialed digit is received. At Line 2 of this function, the variable `command` is the message passed from the calling party to the PBX. For example, the command `A_Dial_0` means that the calling party dials the digit "0." An incomplete list of the Jupiter commands are listed in Table I. For each PBX task, a timeout timer is set. If the task is not complete before the timer expires, some actions are taken to ensure that the PBX can return to an operational state. The command `Time_Out` indicates a timeout timer expiration. The variable `A_Party` represents the telephone number of the calling party, and the variable `B_Party` represents the telephone number of the called party. The `QOverlapSending` function works as follows.

Line 1: The function first stops the `inter_dialed_digit` timeout timer for the calling party. Then one of the three parts in this function is executed.

Part 1: Lines 3–7: This part contains the instructions for the registration procedure. We will describe this part in Section III.

Part 2: Lines 8–24: This part handles the dialed digits sent from the calling party to the PBX. At Line 10, every dialed digit is collected in the buffer `B_Party`. Line 11 sets the next timeout timer for the calling party. The calling party is expected to take the next action before the timer expires. Depending on the number of dialed digits received so far, Lines 12–24 of the `QOverlapSending` function process the dialed digits in two cases.

Case 1: Lines 12–14: The received digits do not provide enough routing information to identify the called party. The

```
   void CallRecord::QOverlapSending(){  /* A_Party is the telephone number of the calling party. */
1        systimeout.cancel(A_Party);    /* B_Party is the telephone number of the called party. */
2        switch (command) {
3              case A_Dial_Cross:
4                if ( one dialed digit has been received ){
5                      remove the dial tone connected to A_Party;
6                      Astate = QFeatures;
7                      systimeout.settime(A_Party);
                 }
                 break;
8              case A_Dial_1: case A_Dial_2: case A_Dial_3: case A_Dial_4: case A_Dial_5:
9              case A_Dial_6: case A_Dial_7: case A_Dial_8: case A_Dial_9: case A_Dial_0: case A_Dial_Star:
10               collect the dialed digit in the buffer B_Party;
11               systimeout.settime(A_Party);
12               if ( only one dialed digit is received ) remove the dial tone of A_Party;
13               else if ( the received digits do not provide enough routing information )
14                     return; /* continue  to receive the next digits */
15               else{ /* the received digits provide enough routing information */
16                     if  ( B_Party represents a handset number ){
17                           CurrentAddress = ReadRoamingTable(B_Party);
18                           if ( the CurrentAddress indicates that the handset is in this WPBX ){
                                   /* intra call routing: */
19                                 digitAnalysis(B_Party);
20                           } else{  /* inter call routing: the CurrentAddress indicates that
                                         the handset is in a remote WPBX */
21                                 SetupTrunk(B_Party,CurrentAddress);
22                           }
23                           break;
24                     } else  digitAnalysis(B_Party);
                 }
                 break;
25             case A_ForwardDis:
26             case Time_Out:
27               Release_PBX_resource(A_Party);
28               Astate=QNull;
                 break;
         }
   }
```

Fig. 3. The `QOverlapSending` function in C++-like description.

TABLE I
AN INCOMPLETE LIST OF THE JUPITER COMMANDS

Command	Description
A_Dial_Cross	The calling party dials the digit "#".
A_Dial_Star	The calling party dials the digit "*".
A_Dial_0	The calling party dials the digit "0".
⋮	⋮
A_Dial_9	The calling party dials the digit "9".
A_ForwardDis	The calling party hangs up the phone.
Time_Out	Timeout timer expires.

control exits the QOverlapSending function and the PBX continues to wait for more dialed digits. If only one digit is received, the dial tone for the calling party is removed at Line 12.

Case 2: Lines 15–24: The received digits provide enough routing information. In this case, the number of received digits stored in B_Party specifies the address of the called party. If the called party is a handset, the call setup procedure for the handset is executed (Lines 16–23). We will describe this part in Section IV. If the called party is a fixed telephone line user, then the digitAnalysis(B_Party) function is invoked to set up the call path from the calling party to the called party. In the digitAnalysis function, if the called party is free, the calling party's state *Astate* is set to QOutgoingCallPro to reflect the transition in BCSM. If the called party is busy or does not exist, *Astate* is set to QNull to reflect a call setup failure.

Part 3: Lines 25–28: If the calling party hangs up the phone at the QOverlapSending state, an A_ForwardDis command is sent from the calling party to the PBX. If the inter_dialed_digit timeout timer for the calling party expires, a Time_Out command is issued to the PBX. In either case, the PBX resources (e.g., DTMF receiver and pulse code modulation stream) for the calling party are released, and *Astate* is set to QNull. Call setup is terminated at this moment.

Step III—Called Party Off-Hook: If the called party picks up the phone, then Astate is set to QActive. The calling and called parties start conversation.

Step IV—Termination of the Conversation: In Jupiter, the either-control model is adopted to terminate the conversation. In this model, termination of a call conversation is controlled by either the calling party or the called party. If the calling party hangs up the phone first, *Astate* is set to QNull and the conversation is terminated. If the called party hangs up the phone first, *Astate* is set to QReleaseReq, and the PBX alarms the calling party to hang up the phone by sending a warning tone. A timeout timer is set, and the calling party is expected to hang up the phone. After the timeout timer expires or the calling party hangs up the phone, *Astate* is set to QNull to free the telephone circuit.

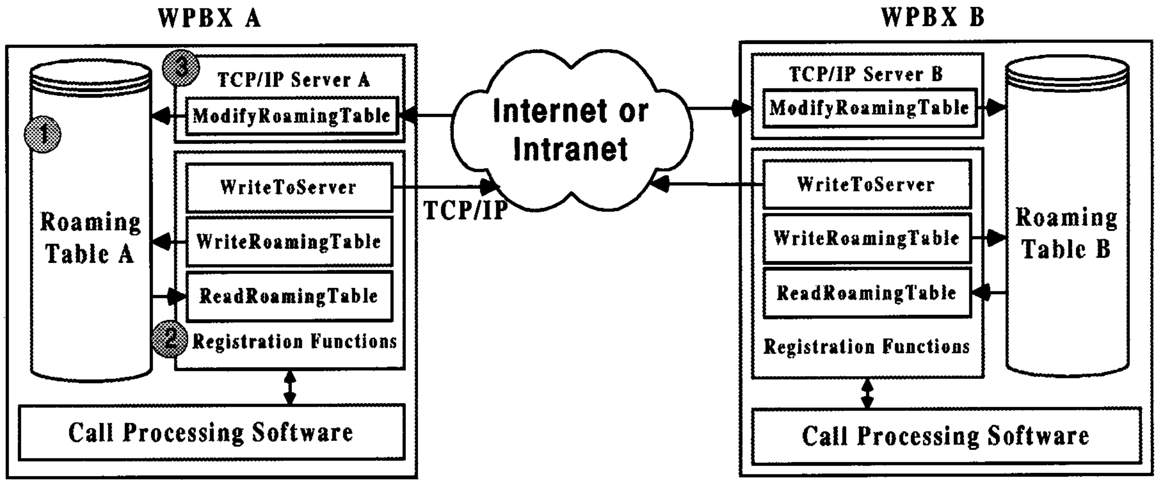

Fig. 4. Mobility management mechanism for WPBX.

Telephone No.	Current Location	other service-related fields
900	02	(omitted)

Fig. 5. Mobile handset record in the roaming table.

III. Registration Mechanism

To implement the registration mechanism for a network of WPBXs, every WPBX is associated with a roaming table as shown in Fig. 4 (1). The WPBX software is modified to include the registration functions [see Fig. 4 (2)] that access a remote roaming table through a TCP/IP server [see Fig. 4 (3)]. In our implementation, a handset has a unique telephone number, which allows the user to attach to different WPBXs without identification ambiguity. Unique telephone numbers can be easily arranged through the PBX numbering plan.

The roaming table is a database that records the current location of a mobile handset. The record consists of a `TelephoneNumber` field to indicate the mobile telephone number, and a `CurrentLocation` field to indicate the routing address of the WPBX where the handset resides. Other service-related fields specify the services related to the mobile handset. Fig. 5 shows a mobile handset record in the roaming table, which indicates that handset 900 resides at WPBX 02. In the above example, a two-digit WPBX address is used just for demonstration purposes. In reality, the length for WPBX address can be extended to provide complete addressing for the WPBXs.

Three registration functions are defined in our implementation:

- The `WriteRoamingTable(TelNo)` function is invoked by a WPBX A to modify A's roaming table when an incoming handset (that moves into WPBX A) makes a registration request to A. The input argument `TelNo` represents the telephone number of the handset, which is used to search the roaming table. When the record is found, the current location field is set to A's routing address.
- The `ReadRoamingTable(TelNo)` function is invoked by a WPBX to retrieve the current location of a handset when a call termination to the handset arrives at that WPBX. The argument `TelNo` is used to search the roaming table, and the corresponding current-location value is returned.
- The `WriteToServer(TelNo, CurrentLocation)` function broadcasts a location update message to TCP/IP servers of all remote WPBXs through the Internet or Intranet. The location update message is of the format ⟨`TelNo, CurrentLocation`⟩.

The TCP/IP server continuously monitors the incoming location update messages sent from remote WPBXs. As mentioned before, a remote WPBX invokes the `WriteToServer` function to issue the location update message. When a TCP/IP server B receives the message ⟨`TelNo, CurrentLocation`⟩, it invokes the `ModifyRoamingTable` function. This function uses `TelNo` to search the roaming table B. When the record is found, the current location field is set to `CurrentLocation`. Note that the location update messages are delivered by using *socket* through the TCP/IP protocol that provides *Reliable Stream Transportation* (RST) [2]. With RST, the location update messages are guaranteed to arrive at the destinations.

Although mobility management is functionally distinguished from call processing, we note that the registration procedure can be integrated with the call processing mechanism by reusing the call setup procedure.

When a handset enters a new WPBX area, it sends a registration message to the base station (BS) through the air interface. The BS sends the WPBX a special sequence "#03∗." We consider the standard DECT [12] approach where the BS-WPBX connection implies the handset number. For other PCS systems such as GSM [6], a signaling message sent from the BS should also indicate the handset number. To accommodate handset registration, we modify the BCSM in Fig. 2 by creating two new states: `QFeatures` and `QRegistration`. The modified call switching model is illustrated in Fig. 6.

- When the digit "#" is received at the `QOverlapSending` state, Lines 3–7 of the `QOverlapSending` function (see Fig. 3) are invoked to process the command `A_Dial_Cross`. At Line 4, if the received digit "#" is the first dialed digit, it means that the BS attempts to set up services (e.g., handset registration), and Lines 5–7 are executed. The dial tone is removed, *Astate* is set to

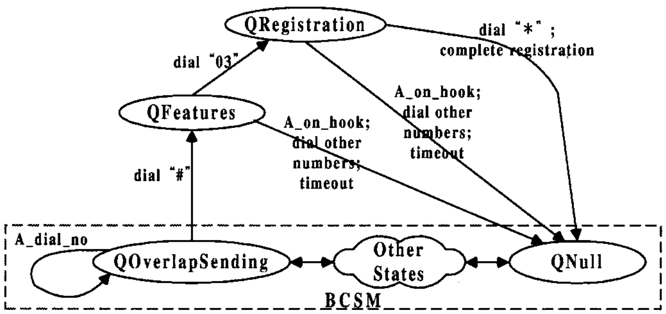

Fig. 6. The modified calling party call model state diagram for registration.

```
 void CallRecord::QFeatures(){
1       systimeout.cancel(A_Party);
2       switch (command){
3           case A_Dial_1: case A_Dial_2: case A_Dial_3:
4           case A_Dial_4: case A_Dial_5: case A_Dial_6:
5           case A_Dial_7: case A_Dial_8: case A_Dial_9:
6           case A_Dial_0: case A_Dial_Cross: case A_Dial_Star:
7               collect the dialed digit;
8               systimeout.settime(A_Party);
9               if( two digits have been collected ){
10                  if ( the received digits are '03' ){
11                      Astate = QRegistration;
12                      break;
13                  } else{
14                      /* set up other services (omitted) */
15                      break;
                    }
                }
                break;
16          case A_ForwardDis:
17          case Time_Out:
18              Release_PBX_resource(A_Party);
19              Astate=QNull;
                break;
        }
 }
```

Fig. 7. The `QFeatures` function in C++-like description.

QFeatures, and the inter_dialed_digit timeout timer is set for the BS. The BS is expected to send the next digit before the timer expires.

- When the WPBX receives the next dialed digit at the QFeatures state, the QFeatures function in Fig. 7 is invoked. The function first stops the inter_dialed_digit timeout timer. Then one of the following two parts is executed.

Part 1: Lines 3–15 handle the transition from state QFeaturs to state QRegistration. The dialed digit is collected at Line 7. The inter_dialed_digit timeout timer is set at Line 8, and the BS is expected to send the next dialed digit before the timer expires. At Line 9, if two dialed digits have been collected, then the WPBX decodes the digits. Lines 10-12 process the service code "03," where a registration request from the handset is identified, and *Astate* is set to QRegistration. Lines 13-15 process services other than handset registration. The details are omitted.

Part 2: Lines 16–19 handle the case when the BS terminates the communication before the registration procedure is complete; i.e., it sends an on-hook signal to the WPBX at the QFeatures state, or the inter_dialed_digit timeout timer expires. The actions taken are exactly the same as Lines 25–28 in the QOverlapSendingfunction.

```
 void CallRecord::QRegistration(){
1   systimeout.cancel(A_Party);
2   switch (command){
3       case A_Dial_Star:
4           WriteRoamingTable(A_Party);
5           WriteToServer(A_Party,the routing address
                of the current WPBX);
6           Release_PBX_resource(A_Party);
7           Astate = QNull;
            break;
8       default:
9           Release_PBX_resource(A_Party);
10          Astate = QNull;
            break;
    }
 }
```

Fig. 8. The `QRegistration` function in C++-like description.

- If the WPBX receives the next dialed digit at state QRegistration, then the QRegistration function in Fig. 8 is invoked. The function first stops the inter_dialed_digit timeout timer. Then the following actions are taken.

Lines 3–6: If the WPBX receives the dialed digit "$*$," then the WriteRoamingTable function is invoked to modify the handset record in the local roaming table, and the WriteToServer function is invoked to modify the handset records in the remote roaming tables. At this point, the registration procedure is completed. The PBX resource for the BS is released, and *Astate* is set to QNull.

Lines 7–8: At the QRegistration state, if a dialed digit other than "$*$" is received, the inter_dialed_digit timeout timer expires, or the BS terminates the communication, the PBX resource for the BS is released, *Astate* is set to QNull, and the registration request is considered as a failure.

Note that the registration messages delivered among the WPBXs are transmitted through the Internet or Intranet without involving the PSTN.

IV. Handset Call Routing Mechanism

Suppose that a call termination to handset p arrives at WPBX A. WPBX A queries its roaming table to identify the WPBX B where p resides, and sets up the trunk to WPBX B (if B is different from A). This call termination mechanism for handset is implemented in Lines 16–23 of the QOverlapSending function (see Fig. 3). At Line 17, the ReadRoamingTable function is invoked to retrieve the routing address of WPBX B (where p resides). The returned value of the ReadRoamingTable function is stored in the variable CurrentAddress. Depending on the value of CurrentAddress, Lines 18–22 of the QOverlapSendingfunction set up the call for handset p in two cases.

Case 1: Intra-PBX Call Routing (Lines 18 and 19): The CurrentAddress value indicates that A and B are the same WPBX. The digitAnalysis function is invoked to set up the call path from the calling party to p (i.e., B_Party). The call setup procedure follows BCSM described in Section II.

Case 2: Inter-PBX Call Routing (Line 21): The CurrentAddress value indicates that WPBXs A and B are different. The SetupTrunk(B_Party, CurrentAddress)

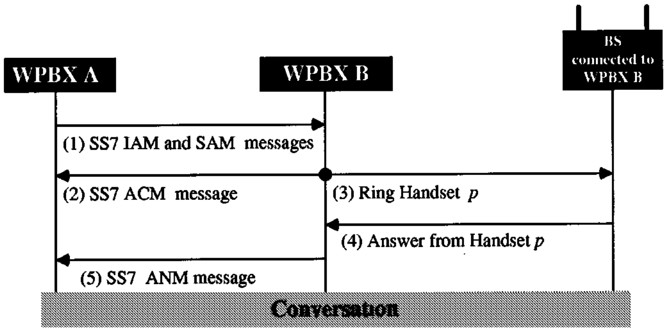

Fig. 9. The message flow for inter-PBX call routing.

function is invoked to set up the trunk between WPBXs A and B. Fig. 9 illustrates the message flow for this function. This function sends the SS7 Initial Address Message (IAM) and Subsequent Address Message (SAM) [11] to provide routing information of handset p to WPBX B [see (1) in Fig. 9]. WPBX B sets up the circuit to p based on the received routing information. Then it sends an SS7 Address Complete Message (ACM) to WPBX A [see (2) in Fig. 9], and starts to ring handset p [see (3) in Fig. 9]. When WPBX A receives the ACM message, *Astate* is set to `QOutgoingCallPro`. If handset p in WPBX B answers the call [see (4) in Fig. 9], WPBX B sends an SS7 Answer Message (ANM) to WPBX A [see (5) in Fig. 9]. *Astate* is set to `QActive`, and the calling party and the called party (handset p) starts the conversation.

We note that the SS7 messages described in **Case 2** are delivered through PSTN.

V. Performance Issue

The delay of Internet signaling affects the registration procedure. The roaming table may provide obsolete information during the transition of location update. Consider a handset p moving from the service area of WPBX A to the service area of WPBX B. Handset p makes a registration request to WPBX B. When WPBX B receives the registration request, it sends the location update messages to other WPBXs in the system as described in Section III. Let T be the period between when the location update message is sent from WPBX B and when the message arrives at a WPBX C. During T, any ps call terminations (i.e., any call terminations to p) at WPBX C will be routed to a wrong WPBX since the roaming table in WPBX C has not been modified. These call terminations are misrouted. It is important to see how the delay of the Internet signaling affects misrouting of call terminations to p.

In the Appendix, we derive the probability $\Pr[K=k]$ that k call terminations to p in a WPBX during period T as

$$\Pr[K=k] = \binom{\frac{1}{(\mu\sigma)^2}+k-1}{k}(1-x)^{1/(\mu\sigma)^2}x^k \quad (1)$$

where

$$0 < x = \frac{\mu^2\sigma^2\lambda}{\mu+\mu^2\sigma^2\lambda} < 1.$$

Assume that the call-waiting service is available, which allows a user to answer other calls while already engaged in a call. In practice, a user cannot simultaneously connect to too many calls. In a typical telephone network, a user can be engaged in, at most, two calls. We consider the conservative situation (that is unfavorable for our approach) where a user may be engaged in three calls. In this scenario, the person may have, at most, three misrouted calls during T if T is reasonably small (e.g., less than 3 minutes). Suppose that there are N WPBXs in the system. Let $\Pr[M=k]$ be the probability that p has k misrouted call terminations, where $k \le 3$. Among the N WPBXs, call terminations at WPBX B (where p moves to) are always routed to p correctly. Thus, if we exclude the traffic to WPBX B, then

$$\Pr[M=0] = (\Pr[K=0])^{N-1}. \quad (2)$$

Similarly,

$$\Pr[M=1] = \binom{N-1}{1}\Pr[K=1](\Pr[K=0])^{N-2} \quad (3)$$

$$\Pr[M=2] = \binom{N-1}{2}(\Pr[K=1])^2(\Pr[K=0])^{N-3} + \binom{N-1}{1}\Pr[K=2](\Pr[K=0])^{N-2} \quad (4)$$

and

$$\Pr[M\ge 3] = \Pr[M=3] = 1-\sum_{i=0}^{2}\Pr[M=i]. \quad (5)$$

Since the call termination traffic of p at a WPBX is a Poisson process with rate λ, the net call termination traffic to p is $\lambda^* = N\lambda$.

We investigate the effects of the net call termination rate λ^*, the standard deviation σ of message sending delay, and the number N of WPBXs on the performance (i.e., $\Pr[M]$), where λ^* and σ are normalized by μ. For example, if the expected value of T is $1/\mu = 2.4$ seconds (i.e., the average TCP/IP signaling message delay is 2.4 seconds), then $\lambda^* = 0.001\mu$ indicates that there is one call termination per 40 minutes to p in the system. We consider two cases. In the first case, the WPBXs in various cities are connected through an uncongested Internet network (or an Intranet network). In this case, the average TCP/IP signaling message delay is less than 2.4 seconds. The call frequency to p is one call termination per 40 minutes (i.e., $\lambda^* = 0.001\mu$). In the second case, the WPBXs are connected through a congested Internet environment, where the average TCP/IP signaling message delay is 24 seconds. Like the first case, the call frequency to p is one call termination per 40 minutes (i.e., $\lambda^* = 0.01\mu$). Based on (2)–(5), we observe the following phenomena.

Effect of λ^*: Figs. 10 and 11 plot $\Pr[M=k]$ $(0\le k\le 3)$ for various N values, where $\lambda^* = 0.01\mu$ and 0.001μ, respectively. In all cases considered in our study, $\Pr[M=0] > 99.1\%$ for $\lambda^* = 0.01\mu$ [Fig. 10(a)] and $\Pr[M=0] > 99.9\%$ for $\lambda^* = 0.001\mu$ [Fig. 11(a)]. In other words, in an uncongested Internet transmission environment (i.e., $\lambda^* = 0.001\mu$), the misrouting probability can be ignored (which is less than 0.1%). In a congested Internet transmission environment (i.e., $\lambda^* = 0.01\mu$), the misrouting probability is reasonably small (which is less than 1%). Furthermore, $\Pr[M=2]$ and $\Pr[M=3]$ are in the order of 10^{-5} [Fig. 10(c)] and 10^{-7} [Fig. 10(d)] for $\lambda^* = 0.01\mu$,

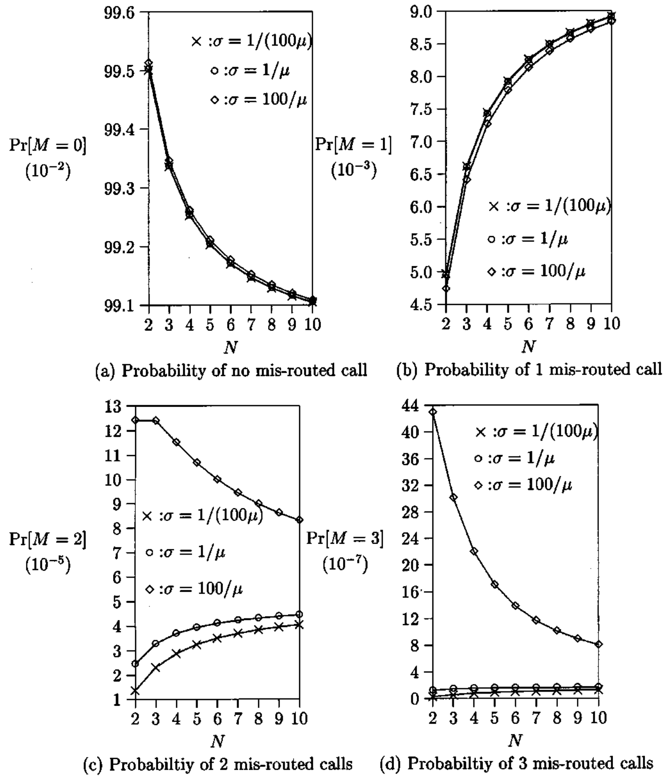

Fig. 10. Effects of N and σ on $\Pr[M]$ ($\lambda^* = 0.01\mu$).

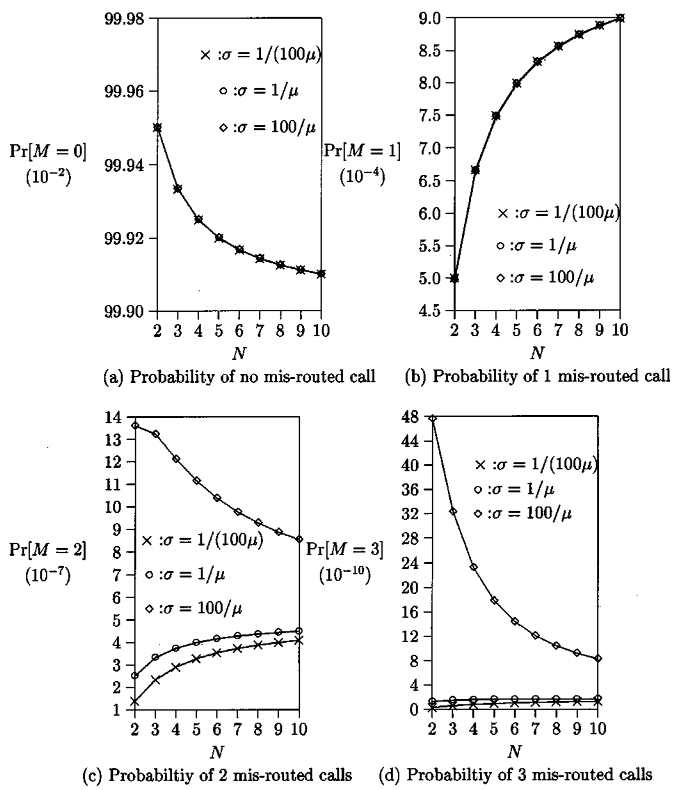

Fig. 11. Effects of N and σ on $\Pr[M]$ ($\lambda^* = 0.001\mu$).

respectively. For $\lambda^* = 0.001\mu$, $\Pr[M=2]$ and $\Pr[M=3]$ are in the order of 10^{-7} [Fig. 11(c)] and 10^{-10} [Fig. 11(d)], respectively. Thus, the misrouted probabilities for multiple calls can be ignored in all cases.

Effect of N: Figs. 10(a) and 11(a) indicate that $\Pr[M=0]$ decreases as N increases. This phenomenon is due to the fact that for a fixed λ^*, the misrouted traffic to the network [which is $(N-1/N)\lambda^*$] increases as N increases. Even if the net misrouting traffic is fixed, the stochastic phenomenon [10] implies that with a fixed net arrival rate, if there are more arrival sources (i.e., more number of WPBXs), it is more likely to observe arrivals during a fixed period. Thus, adding more WPBXs in the network will degrade the performance of the network. However, this effect is not significant. When $N=9$, by adding one more WPBX, $\Pr[M=0]$ is degraded by 0.011% (for $\lambda^* = 0.01\mu$) and 0.0012% (for $\lambda^* = 0.001\mu$). On the other hand, when $N=2$, by adding one more WPBX, $\Pr[M=0]$ is degraded by 0.166% (for $\lambda^* = 0.01\mu$) and 0.0166% (for $\lambda^* = 0.001\mu$). To conclude, the performance degradation due to adding extra WPBXs can be ignored.

Effect of σ: The standard deviation σ of message sending delay determines the probability of observing long/short Internet delay times. As σ increases, more long and short Internet delay times are observed. More short and long Internet delay periods imply larger $\Pr[M=0]$ and $\Pr[M=3]$. This analysis is consistent with our observation in Figs. 10 and 11 which indicate that as σ increases, $\Pr[M=0]$, $\Pr[M=2]$, and $\Pr[M=3]$ increase, and $\Pr[M=1]$ decreases. With a small σ, $\Pr[M=2]$ and $\Pr[M=3]$ increase as N increases. On the other hand, this misrouting probability decreases as N increases for large σ. This nonintuitive phenomenon is due to complicate interaction among input parameters that we cannot explain. Since $\Pr[M=2]$ and $\Pr[M=3]$ are very small for large N, the effect of σ can be ignored under the normal operation conditions.

VI. Conclusion

We designed an enterprise mobile phone network based on WPBX. We showed how to modify the PBX call model to accommodate mobility management where the location update messages are transmitted through Internet or Intranet. We have implemented an enterprise mobile phone network using a commercial PBX product called Jupiter. Although the configuration in the current implementation only consists of two WPBXs, the implementation can be easily extended to accommodate more WPBXs.

In our design, it is important to investigate if Internet message delay will cause misrouting of phone calls. In the current implementation with 2 WPBXs, message delay does not cause any misrouting problem. We used an analytic model to study the performance of the system with more than 2 WPBXs in the congested Internet environment. Our study indicated the following.

- In an uncongested Internet transmission environment, the misrouting probability can be ignored (less than 0.1% in our study). For a congested Internet, the misrouting probability is reasonably small (less than 1% in our study).
- The misrouting probabilities for multiple calls can be ignored in all cases (less than 10^{-5} in our study).
- The performance degradation due to adding extra WPBXs can be ignored (the degradation is less than 0.012% in our study).

- When the net call termination traffic to a handset is large, the standard deviation σ of the Internet message delays has significant effect on the misrouting probability.

Our study showed that with a large number of WPBXs and long Internet message delays, the misrouting probability can be limited within 1%. This performance result is considered satisfactory. Furthermore, the misrouted calls are not necessarily lost. A misrouted call can be forwarded to the voice mailbox or the secretary of the called party, and the called party can call back later. To conclude, our design provides mobility management for an enterprise mobile phone network, which utilizes the Internet/Intranet for signaling. Our design does not modify the existing signaling capability of public switched telephone network, and thus results in low costs for implementation and maintenance.

APPENDIX
DERIVATION FOR $\Pr[K = k]$

Suppose that the call terminations at a WPBX are a Possion process with rate λ, and T is a random variable with a Gamma density function $f(t)$ with mean $1/\mu$, standard deviation σ, and the Laplace transform $f^*(s)$ where

$$f^*(s) = \left(\frac{\mu}{\mu + \mu^2\sigma^2 s}\right)^{1/(\mu\sigma)^2}. \tag{6}$$

Note that our derivation applies to any message delay distributions whose Laplace transforms have closed forms. The Gamma distribution is selected to present the message delay distribution because this distribution can be used to approximate many distributions as well as the measured data obtained from the PCS fields [8]. Let random variable K be the number of p's call terminations at a WPBX during period T. The probability that $K = k$ during period $T = t$ is

$$\Pr[K = k \,|\, T = t] = \frac{(\lambda t)^k}{k!}\, e^{-\lambda t}. \tag{7}$$

The derivation for $\Pr[K = k]$ is described as follows.

$$\begin{aligned}
\Pr[K = k] &= \int_{t=0}^{\infty} \Pr[K = k \,|\, T = t] f(t)\, dt \\
&= \left(\frac{\lambda^k}{k!}\right) \int_{t=0}^{\infty} t^k f(t) e^{-\lambda t}\, dt \\
&= \left(\frac{\lambda^k}{k!}\right) \left[(-1)^k \frac{d^k}{ds^k} f^*(s)\right]\Bigg|_{s=\lambda} \\
&= \left(\frac{\lambda^k}{k!}\right) \left[(-1)^k \frac{d^k}{ds^k} \left(\frac{\mu}{\mu + \mu^2\sigma^2 s}\right)^{1/(\mu\sigma)^2}\right]\Bigg|_{s=\lambda} \\
&= \binom{\frac{1}{(\sigma\mu)^2} + k - 1}{k} \left(\frac{\mu}{\mu + \mu^2\sigma^2\lambda}\right)^{1/(\mu\sigma)^2} \\
&\quad \times \left(\frac{\mu^2\sigma^2\lambda}{\mu + \mu^2\sigma^2\lambda}\right)^k \\
&= \binom{\frac{1}{(\mu\sigma)^2} + k - 1}{k} (1 - x)^{1/(\mu\sigma)^2} x^k.
\end{aligned}$$

We validate (1) by showing that the expected value of K is

$$E[K] = E[T]\lambda. \tag{8}$$

From (1), we have

$$\begin{aligned}
E[K] &= \sum_{k=1}^{\infty} k \Pr[K = k] \\
&= (1 - x)^{1/(\mu\sigma)^2} \sum_{k=1}^{\infty} k \binom{\frac{1}{(\mu\sigma)^2} + k - 1}{k} x^k.
\end{aligned} \tag{9}$$

By using the general binomial series summation [9], we have

$$\sum_{k=1}^{\infty} k \binom{\frac{1}{(\mu\sigma)^2} + k - 1}{k} x^k = \frac{\frac{x}{(\mu\sigma)^2}}{(1 - x)^{1 + (1/(\mu\sigma)^2)}}$$

and (9) is rewritten as

$$\begin{aligned}
E[K] &= (1 - x)^{1/(\mu\sigma)^2} \left[\frac{\frac{x}{(\mu\sigma)^2}}{(1 - x)^{1 + (1/(\mu\sigma)^2)}}\right] \\
&= \frac{x}{(\mu\sigma)^2 (1 - x)} \\
&= \frac{\lambda}{\mu}.
\end{aligned} \tag{10}$$

Since $E[T] = 1/\mu$, it is apparent that (10) and (8) are the same. Equation (10) states that the expected number $E[K]$ is independent of the T distribution. However, $\Pr[K]$ is sensitive to the T distribution.

REFERENCES

[1] I. Chlamtac, B. Khasnabish, and Y.-B. Lin, "The wireless segment for enterprise networking," *IEEE Network*, vol. 12, pp. 50–55, July/Aug. 1998.

[2] D. E. Comer, *Interworking with TCP/IP Vol. I: Principles, Protocol, and Architecture*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[3] R. A. Dayem, *PCS and Digital Cellular Technologies*: Prentice-Hall, 1997.

[4] EIA/TIA, "Cellular intersystem operations (Rev. C)," Technical Report, EIA/TIA, 1995.

[5] ETSI/TC, "Mobile application part (MAP) specification, version 4.8.0," Technical Report Recommendation GSM 09.02, ETSI, Aug. 1997.

[6] V. K. Garg and J. E. Wilkes, *Principles & Applications of GSM*: Prentice-Hall, 1999.

[7] ITU-T, "Digital subscriber signaling system no. 1: Network layer," Technical Report Recommendation Q.931, ITU-T, Mar. 1993.

[8] N. L. Johnson, *Continuous Univariate Distributions-1*. New York: Wiley, 1969.

[9] L. B. W. Jolley, *Summation of Series*, 2nd ed: Dover Publications, 1961.

[10] L. Kleinrock, *Queuing Systems: Volume I—Theory*. New York: Wiley, 1976.

[11] Y.-B. Lin and S. K. DeVries, "PCS network signaling using SS7," *IEEE Personal Commun. Mag.*, pp. 44–55, June 1995.

[12] J. Phillips and G. M. Namee, *Personal Wireless Communication with DECT and PWT*: Artech House, 1998.

Phone Lin received the B.S.C.S.I.E. and Ph.D. degrees from National Chiao Tung University, Taiwan, R.O.C., in 1996 and 2001, respectively.

He is currently an Assistant Professor with the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, R.O.C. His current research interests include design and analysis of personal communications services networks, mobile computing, and performance modeling.

Yi-Bing Lin (SM'96) received the B.S.E.E. degree from National Cheng Kung University in 1983, and the Ph.D. degree in computer science from the University of Washington in 1990.

From 1990 to 1995, he was with the Applied Research Area, Bell Communications Research (Bellcore), Morristown, NJ. In 1995, he was appointed as a Professor of the Department of Computer Science and Information Engineering (CSIE), National Chiao Tung University (NCTU). In 1996, he was appointed as Deputy Director of Microelectronics and Information Systems Research Center, NCTU. During 1997–1999, he was elected as Chairman of CSIE, NCTU. His current research interests include design and analysis of personal communications services network, mobile computing, distributed simulation, and performance modeling. He is coauthor with I. Chlamtac of *Wireless and Mobile Network Architecture* (New York: Wiley).

Dr. Lin is an Associate Editor of IEEE NETWORK, an Editor of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Wireless Series, an editor of IEEE PERSONAL COMMUNICATIONS MAGAZINE, and an Editor of numerous other journals; as well as Guest Editor for IEEE TRANSACTIONS ON COMPUTERS Special Issue on Mobile Computing, and a Guest Editor for IEEE COMMUNICATIONS MAGAZINE Special Issue on Active, Programmable, and Mobile Code Networking. He served as Program Chair for the 8th Workshop on Distributed and Parallel Simulation, General Chair for the 9th Workshop on Distributed and Parallel Simulation, and Program Chair for the 2nd International Mobile Computing Conference. He received the 1998 and 2000 Outstanding Research Awards from National Science Council, R.O.C., and 1998 Outstanding Youth Electrical Engineer Award from CIEE, R.O.C. He is an Adjunct Research Fellow of Academia Sinica.