Supporting SIP Personal Mobility for VoIP Services

Tsan-Pin Wang[1] and KauLin Chiu[2]

[1]Department of Computer and Information Science, National Taichung University,
140, Min-Shen Rd, Taichung, 403 Taiwan, R.O.C.
tpwang@ms3.ntctc.edu.tw
[2]Department of Computer Science and Information Engineering, National Chung Cheng University,168, University Rd., Min-Hsiung Chia-Yi, Taiwan, R.O.C.
Kaulin.chiu@msa.hinet.net

Abstract. SIP is promising for VoIP signaling to support personal mobility. In this paper, we introduce and compare single registration (SR) and multiple registration (MR) for personal mobility. The SR scheme can not support personal mobility without user's assistance. In contrast, the MR scheme supports personal mobility inherently using sequential search or pure parallel search. Sequential search may suffer from long delay for call setup, while pure parallel search consumes network resource. To compromise the two schemes, we propose pipelined search for multiple registration.

1 Introduction

In early days, the key technology of VoIP was H.323 [1][2]. The H.323 standard was specified by the ITU-T Study Group 16. The advantages of H.323 include high-reliability and easy to maintain. However, H.323 still has lots of shortcomings, for example, lack of flexibility and high construction cost. Because of these shortcomings, H.323 is not deployed worldwide.

In order to solve these shortcomings the Internet Engineering Task Force (IETF) draws up a standard protocol, Session Initiation Protocol (SIP) [3]. SIP is an application-layer signaling protocol for initiation, modification, and termination of sessions with two or more participants. SIP offers a chance to realize low construction cost and high flexibility. The media stream of SIP can be video, audio or other Internet-based multimedia applications, such as white board, shared text editors, etc.

Unlike H.323, SIP is a text-based protocol similar to Hyper Text Transfer Protocol (HTTP) [4]. SIP and HTTP have a lot of similarity on processing and transmitting information. SIP continues using the request-response model, much of the HTTP syntax, header fields and semantics. Because of its simplicity and popularity, SIP has been promising in VoIP environment [5].

SIP has several key components [6], including user agents, redirect servers, proxy servers and registrars. User Agents (UAs) are endpoint devices that originate and terminate SIP requests (signaling). They can be either clients (UAC) that initiate requests or servers (UAS) that respond to the requests, or more normally a combination of both. The UAs are addressed by SIP-URLs that are similar to the email address form, for example, sip:TPwang@sip.pu.edu.tw or tel: TPwang@sip.pu.edu.tw.

Redirect Servers receive requests and push routing information for requests back in responses to the client. Registrars are special User Agent Servers that handle “REGISTER” requests. SIP users/devices use “REGISTER” requests to dynamically register their current locations. After registration, the SIP user/device can be contacted even when they move.

Typically, UAs will send a “REGISTER” message to a specific registrar server. If username in the “REGISTER” message is authorized, it can receive a final response (200 OK) and the registrar server can store user information to the location database, as shown in Fig. 1.

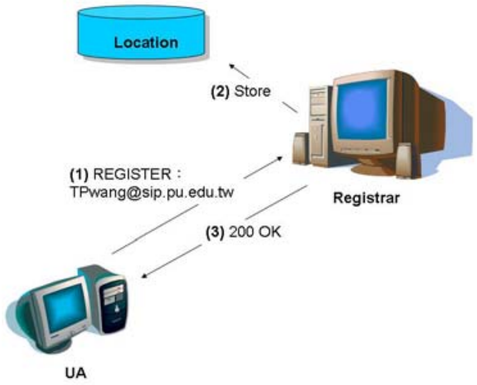


Fig. 1. Registration Scenario

Proxy Servers are elements that route requests to the user agent server and responses to the user agent client. A proxy server can operate in either a stateless proxy or a stateful proxy. A stateless proxy server just simply forwards incoming requests to another server or client, without dealing with any reliability. It forwards every request downstream to a single element determined by making a routing decision based on the request and simply forwards every response it receives upstream. In contrast, a stateful proxy maintains information (specifically, transaction state) about every received request and any responses produced by the request message that it sent.

A stateful proxy can be a *forking proxy* [4] that can route request to multiple destinations. Using forking is useful when proxy servers do not know the exact final destination. Proxy servers can either try a set of destination in pure parallel search, sequential search or other hybrid algorithms.

Practically, we can implement registrar, proxy, redirect server in the same machine, which is called “***Call Server***”.

A successful SIP call invitation mush consists of two messages, an INVITE and followed by an ACK [7] [8]. The INVITE request asks the callee to join a particular conference or to establish a two party conversation. The request message’s body may include some description of the session using Session Description Protocol (SDP). SDP contains distinction address, codec, connection ports and other information. After the callee agrees and answers to join this call, the caller confirms that it has received a "200 OK" response by sending an ACK message. A success response must indicate which media type the callee wishes to receive and may indicate the media callee is going to send. Finally, the media stream will be established by using Real Time Protocol (RTP) and Real Time Control Protocol (RTCP) to transport digital audio or video.

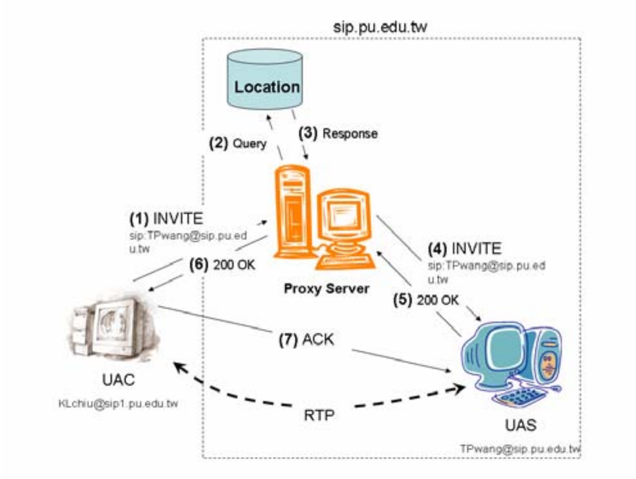


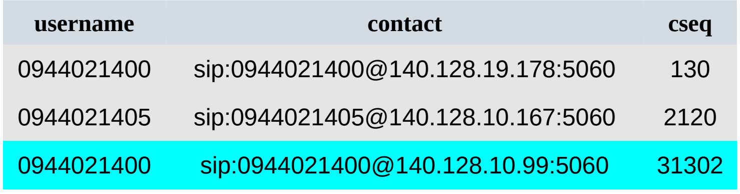
Fig. 2. Session Set-up

Consider an example of the session setup in which an INVITE message is sent from KLchiu@sip1.pu.edu.tw to TPwang@sip.pu.edu.tw, as shown in Fig. 2. Typically, all requests will be sent to a predefined local proxy server. Then the local proxy server would check the registrar's database in order to look up whether the callee is on-line or not. If the callee is found, the proxy server would forward the INVITE message to appropriate UAS. When TPwang answers the call, UAS would send a "200 OK" message to UAC via the proxy server. Finally, this call will be established using RTP protocol.

2 Single and Multiple Registration

In general, most of inter-communication platforms accept their user to register only place at the same time, for instance, MSN messenger and skype. This architecture is referred to as ***single registration (SR)***. The SR architecture does not support personal mobility inherently because the registration cannot be transparent to the user. In other word, it cannot address a single user location at different terminals using the same logical address. This way will be very inhumanity, because we cannot always ask users to sit in front of the computer or hand-carrying terminals. Meanwhile, the proxy server must accept to authorize shorter legal service time in order to alleviate the phenomenon that users have left the terminal. In RFC-3261, the value called Expire is defined to solve this problem. The default value could be 1,800 or 3,600 seconds.

In order to solve the drawback, a good solution is let all terminals of the user can register into registrar server at the same time. This method is called ***multiple registration (MR)***. Fig. 3 demonstrates an example of the contact information stored in the iptel SER's registrar [9] for multiple registration.



username	contact	cseq
0944021400	sip:0944021400@140.128.19.178:5060	130
0944021405	sip:0944021405@140.128.10.167:5060	2120
0944021400	sip:0944021400@140.128.10.99:5060	31302

Fig. 3. iptel SER's registrar for multiple registrations.

Using multiple registration, the forking proxy [10] can search several destinations of the callee. Typically, there are two algorithms to search the current location of the callee: sequential search and pure parallel search. The sequential search will use First In First Service (FIFS) to determine the processing order. In worst case, this method has an important and critical shortcoming that calls setup will fail. That is, when user is near device registered recently. Therefore, a timeout mechanism will be necessary to continue searching the next possible location. In contrast, the forking proxy searches all destinations in parallel. However, the pure parallel search consumes more network resource.

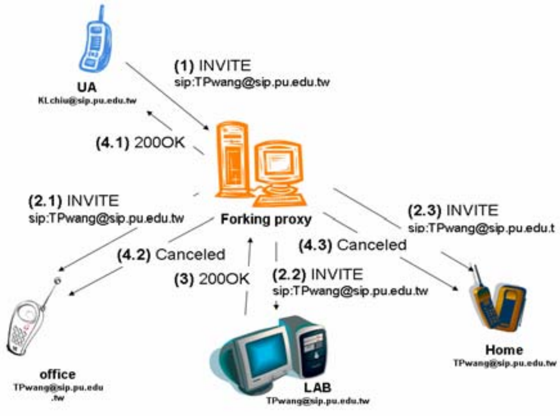


Fig. 4. Pure Parallel Search

Consider an example of the pure parallel search. Suppose that TPwang may move between three locations: LAB, office, and home as shown in Fig. 4. When KLchiu wants to make a phone call to TPwang, the forking proxy will fork three INVITE messages to all of TPwang's possible terminals at the same time if they are on-line. In this example, we assume that TPwang is at laboratory and he answers this call in the LAB. Therefore, the session will be established from KLchiu's UA to TPwang's UA at laboratory. Finally, other INVITEs will be cancelled using CANCEL method.

3 Pipelined Search Algorithm

In this section, we propose a pipelined search scheme for multiple registration. Pipelined search is a hybrid method which combines sequential search and pure parallel search. It can compromise call setup delay and search cost at the same time. This algorithm defines a "d" parameter value which is used to delay the time of issuing the next request according to network status and user's behaviors. In most situations, d-value ranges from several hundred millions to several seconds. And, we use q-value for priority. It is possible to generate the q-values by analyzing the user's mobility behaviors. We also define "Group" for those sent together. Group members will have the same or similar q-value. The group concept is shown in Fig. 5.

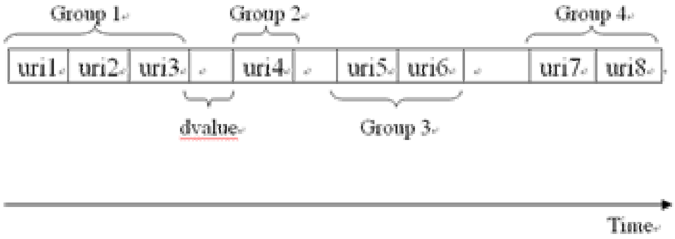


Fig. 5. The group concept

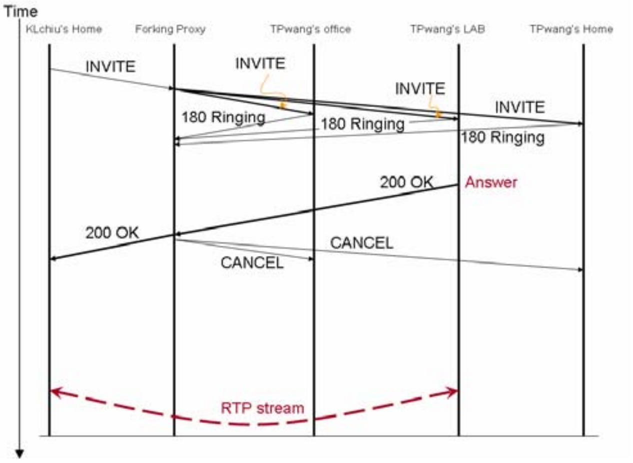


Fig. 6. Timing diagram for parallel search

We can get a priority list by calculating q-values and regulating the size of d-values to determine the way of search. When d-value is large, pipelined search is similar to the sequential search. On the other hand, pipelined search is similar to pure parallel search when d-value is small.

For simplicity, in Fig. 6 and Fig. 7, we ignore some provisional response messages, and "183 Call Pregree" messages between TPwang and KLchiu's UA. When using pure parallel search algorithm, the forking proxy will receive one "INVITE", three "180 Ringing", one "200 OK", three "ACK" and forward three "INVITE", one "180 Ringing, one "200 OK", 2 "CANCEL", one "ACK". Therefore, we can derive the total number of messages sent or processed by the forking proxy as the following equation (1).

Consider the omitted provisional messages, the parallel search method will waste more resource of the network and search cost. In the piplelined search case, the phone call will be bulit as soon as possible beause of TPwang is in the LAB as shown in Fig. 7. So, forking proxy does not send the third "INVITE" to TPwang's Home and can reduce the number of sent messages.

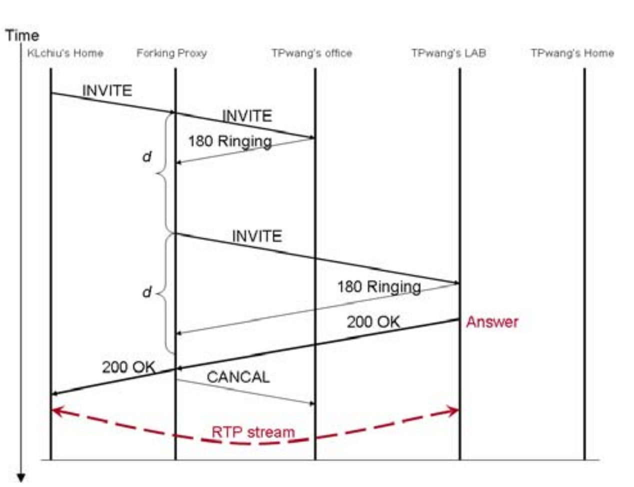


Fig. 7. Timing diagram for pipelined search

$$f(N) = 4N + 4 \tag{1}$$

where N is the number of terminals.

Moreover, if the forking proxy chooses higher q-value to send the "INVITE" message first and delays a time period of d for the subsequent "INVITE" messages. The search cost will be significantly reduced.

4 Performance Analysis and Comparison

In this section, we first compare the performance comparison of single registration and multiple registration mechanisms, and then discuss the impact of locality on multiple registration with pipeline parallel search.

4.1 Comparison of Single and Multiple Registrations

In the literature, perfornance evaulation of Mobile IP and SIP can be found in [11][12]. However, there is no research to compare single and multiple registrations for SIP personal mobility. In this subsection, we analyze and compare the performance of single registration and multiple registration.

Single registration is suitable for users with high Call-to-Mobility Ratio (CMR) and without locality behavior. It means the SR scheme is suitable for terminal mobility instead of personal mobility beause a terminal will always update its location information automatically in order to keep the lastest current position when users move from one place to another. However, supporting personal mobility in SR scheme relies on user to assist UA to send "REGISTER" message to the registrar server. If a user moves and forgets to register in the new terminal, a call to the user will fail to be delivered.

Multiple registration is suitable for users with low CMR and with regular mobility pattern with locality behavior. If the user's mobility pattern is regular, the MR scheme allows the SIP terminals to ask the forking proxy issuing a longer legal service time in the "REGISTER" message, for instance, 7200 seconds or more. In the best case, registration is necessary only in the first access. Since registration is unusual in MR scheme, the cost of registration will tend to be ignorable in the long term.

Sequential search, pure parallel search, and pipelined search can be used to search the user's current location for multiple registration. Sequential search suffers from longer delay to wait for timeout on searching the possible user location. It is unsuitable for the caller without patience to wait the long delay. On the other hand, the pure parallel search will outperform others in terms of its short delay for call setup. Because all INVITE messages will be sent at the same time, this algorithm is suitable only for the user with locality in a small number of possible locations. Otherwise, many network resources will be waste. Note that the performance of pipelined search depends on the distribution of user mobility pattern.

In the following, we derive the total cost for SR and MR schemes. In general, the total cost of a scheme is the sum of the paging/searching cost and the registration cost. The paging cost indicates the number of messages that a proxy spends for searching the user location. And, the registration cost is the messages sent to register the user location. Normally, every paging consists of eight incoming or out-coming messages that include INVITE, 100 trying, 180 ringing and 200 OK. Note that the provisional messages (100 trying and 180 ringing) are omitted in Fig.2 for simplicity. On the other hand, the registration includes two messages that are "REGISTER" and "200 OK" as shown in Fig.1.

Suppose that the call rate is λ and the mobility rate is μ for a SIP user. That is, a proxy server may perform λ times paging and receive μ times registration in a time unit. We derive two equations for the total cost of single registration and multiple registration as follows.

According to the above description, the total cost of single registration (Cost_S) is

$$Cost\_S = 8*\lambda + 2*\mu \tag{2}$$

As we mentioned above, the cost of registration will tend to be ignorable in the long term. Therefore, the total cost of multiple registration (Cost_M) is equal to the paging cost. From Equation (1), the total cost is

$$\begin{aligned} Cost\_M &= \lambda * [4N + 4] + 0 \\ &= 4N\lambda + 4\lambda \end{aligned} \qquad (3)$$

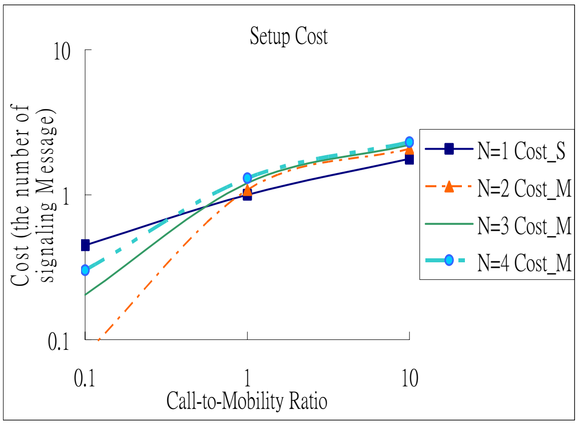


Fig. 8. The impact of CMR to call setup cost

In equation (3), we assume that “ACK” message will pass through the proxy server for stateful processing.

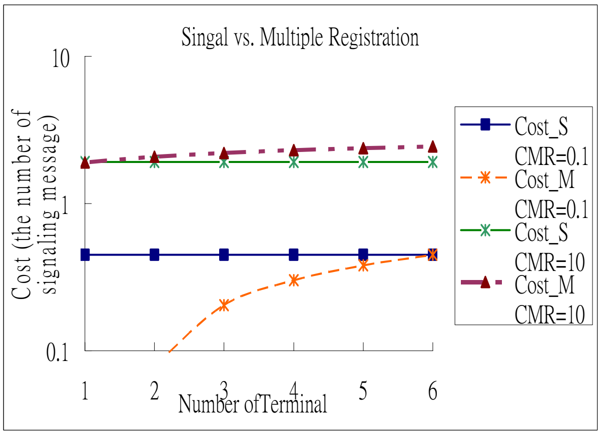


Fig. 9. The impact of n to call setup cost

The results derived from equations (2) and (3), are shown in Fig. 8 and Fig. 9. In order to demonstrate the major difference, we perform nature logarithm treatment to the total cost. Fig. 8 shows the impact of CMR on the total call setup cost. Note that the call setup cost increases as CMR increases. Fig. 9 shows the impact of the number of terminal to the total call setup cost. Single registration accepts only one terminal registering to the registrar at the same time. Consequently, its setup cost is always the same. It is obvious that the costs for single and multiple registrations have only little difference when the user has only two terminals. However, single registration must issues “REGISTER” request message. Therefore, its cost is higher than that of multiple registration.

4.2 Impact of Locality on Pipeline Search

After comparing MR with SR, we further discuss the impact of locality behavior on pipelined search in multiple registration based environment. Performance metrics include the mean call-setup delay and the mean number of message sent to setup a call. In addition, we consider two mobility patterns: uniform and locality distributions. Uniform distribution means the user appears uniformly in all possible locations. In contrast, locality distribution means that the user may appear in a few locations with higher probability.

Prior to deriving the results, we list the used notation as follows.

RT：Response time
d：dvalue
Pi：Probability value
N：The number of terminals
t：The time of successful setting up a call

We first derive the mean delay time (t) for setting up a call.

Uniform distribution

Since the user may appear uniformly in all possible locations, the probability of the user in each location is the same. Therefore, the mean delay for call setup is

$$E(t) = \sum_{i=1}^{N} [RT + (i-1) * d] * p_i$$

$$\because\ p_n = p_1 = p_2 = \ldots = \frac{1}{N}$$

$$\therefore\ E(t) = RT + \frac{N-1}{2} * d \qquad (4)$$

Locality distribution

Without loss of generality, we assume that the probability (pBiB) of the user in location i is twice the probability in location i+1 (pBi+BB1B) for all possible i in locality distribution. Consequently, we can easily derive the probability pBiBB.B and the mean call setup delay as follows.

$$C = 1 + 2 + 4 + 8 + \ldots + 2^{N} = 2^{n} - 1$$

$$p_{i} = \frac{2^{N-i}}{2^{N} - 1} \qquad 1 <= i <= N$$

$$E(t) = \sum_{i=1}^{N} [RT + (i-1) * d] * p_{i}$$

$$= RT + \frac{d}{2^{N} - 1} * \sum_{i=1}^{N-1} i * 2^{N-i-1} \tag{5}$$

In Fig. 10, we assume the value of RT is 3. The values in the pure parallel search are not changed. Because pure parallel search always sends to all destinations at the same time, it produces no extra delay. And, the curve with uniform distribution will grow up linearly according to the number of group. In contrast, the curve with locality distribution will lead to a fix value when N approaches 10. Note that the difference between different d values is very small.

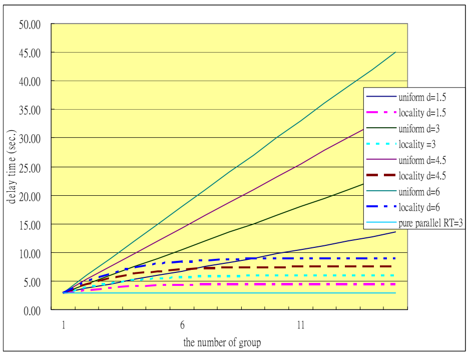


Fig. 10. The Delay Time for Invitation Call

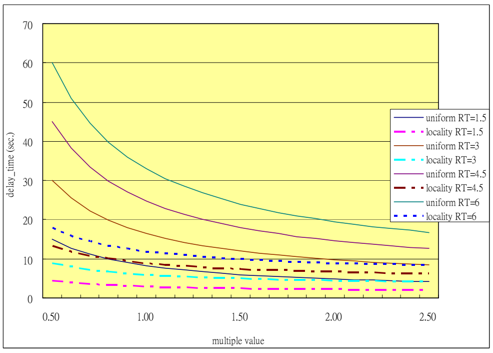


Fig. 11. The Impact of the M to Call Setup

In Fig. 11, we define the value of M as RT divided by d and N is equal to 10. The result from this figure is very close to that of pure parallel search when M is more and

more larger (i.e., RT is more than d) and the delay time will be reduced. In any cases, the one with locality distribution outperforms the one with uniform distribution.
In the following, we derive the mean number of messages sent for call setup. Let n denote the number of messages sent for setting up a call.

Uniform distribution

$$
\begin{aligned}
E(n) &= \sum_{i=1}^{N} p_i * [(i-1) + \left\lceil \frac{RT}{d} \right\rceil \\
&= \frac{N+1}{2} - 1 + \left\lceil \frac{RT}{d} \right\rceil
\end{aligned}
\qquad \textbf{(6)}
$$

Locality distribution

$$
\begin{aligned}
E(n) &= \sum_{i=1}^{N} p_i * [(i-1) + \left\lceil \frac{RT}{d} \right\rceil \\
&= \frac{1}{2^N - 1} \sum i * (2^{N-i}) - 1 + \left\lceil \frac{RT}{d} \right\rceil
\end{aligned}
\qquad \textbf{(7)}
$$

In Fig. 11, the locality distribution approaches to a fixed value when N is near to 10. Regardless of the paging cost (means sent messages) with uniform or locality distribution, pipeline search performs better than pure parallel search.

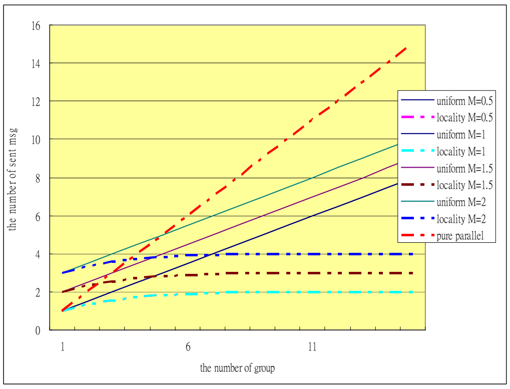


Fig. 12. Impact of the number of group

According to the above results, we observe that the one with locality distribution is very stable when N approaches to 10. It is strongly recommended that the system should provide multiple registration service for the users with locality behavior. In other hand, RT is very difficult to control in different network environment. Network manager can refer to our results (in figures 11 and 12) to adjust the d-value.

5 Conclusions

In this paper, we compare the performance of single registration and multiple registration. The single registration scheme is suitable for users with high call to mobility ratio. In contrast, multiple registration is suitable for users with low call to mobility ratio. Moreover, sequential search suffers from long setup delay, while pure parallel search consumes more resource to paging the user location. A compromise solution we propose is pipelined search. In pipeline search, if we can get a good algorithm for deriving d-value and q-value, the pipelined search would reduce wasted resource and improve system performance.

Acknowledgement

This work was supported in part by NSC under Grants NSC93-2219-E-126-001, NSC 93-2213-E-126-002, and NSC 94-2213-E-126-006, Taiwan, R.O.C.

Reference

1. ITU-T Recommendation H.323, “Packet-based multimedia communication systems”, 1998.
2. B. Vlaovic, Z. Brezocnik, “Packet based telephony”, EUROCON'2001, Trends in Communications, International Conference on., Vol. 1 , pp. 210 – 213, vol.14-7, July 2001
3. SIP Tutorial, at http://www.iptel.org/sip/ (last visit 2 September 2004)
4. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, “SIP: Session Initiation Protocol”, RFC 3261, June 2002
5. K. Daniel Wong, Ashutosh Dutta, Jim Burns, Ravi Jain, Kenneth Young “A multilayered mobility management scheme for auto-configured wireless IP networks”, IEEE Wireless Communications, Vol. 10, pp.62-69, Oct. 2003
6. H. Schulzrinne, E. Wedlund, “Application layer mobility using SIP”,Mobile Computing and Communications Review, Volume 4, Number 3
7. E. Wedlund, H. Schulzrinne, “Mobility support using SIP”, 2ndACM/IEEE International Conference on Wireless and Mobile Multimedia, Seattle, Washington, Aug. 1999
8. M. Handley, V. Jacobson “SDP: Session Description Protocol”, RFC2327, April 1998
9. http://www.iptel.org/
10. Henry Sinnreich, Alan B. Johnston, “Internet Communications Using SIP”, John Wiley & Sons, Inc., New York, NY, USA
11. Jin-Woo Jung, Hyun-Kook Kahng, Ranganathan Mudumbai, Doug Montgomery, “Performance Evaluation of Two Layered Mobility Management using Mobile IP and Session Initiation Protocol”, at http://w3.antd.nist.gov/pubs/sip-mip-jwjung-globecom 2003.pdf
12. Peng Sun, Sam Y. Sung, Zhao Li, “Performance Evaluation and Analysis of Protocols for IP Mobility Support: A Quantitative Study”, http://www.iscs.nus.edu.sg/~ssung/publications /1011.pdf