Preprint: IEEE Infocom, Mar. 21-25 1999.

Enhancing Survivability of Mobile Internet Access Using Mobile IP with Location Registers

Ravi Jain, Thomas Raleigh, Danny Yang, Li Fung Chang*
Applied Research, Bellcore

Charles Graff, Michael Bereschinsky, Mitesh Patel
U. S. Army CECOM

Abstract. The Mobile IP (MIP) protocol for IP version 4 provides continuous Internet connectivity to mobile hosts. However, currently it has some drawbacks in the areas of survivability, performance, interoperability with protocols for providing QoS. We have proposed an alternative protocol, *Mobile IP with Location Registers (MIP-LR)*, which overcomes some of these drawbacks and is closer to the "service node" database approach used in the Public Switched Telephone Network (PSTN): before launching a packet to the mobile host, the sender first queries a database, called the Home Location Register (HLR), to obtain the recipient's current location. MIP-LR is designed for operation in enterprise environments or within logical administrative domains.

In this paper we focus on showing how MIP-LR enhances the survivability of MIP by eliminating some of the functions which MIP introduces for mobility support, allowing the HLR to be placed outside the mobile's home network in case the latter is particularly vulnerable, and replicating and distributing HLRs. We present two schemes for managing the multiple HLRs and enabling mobile and correspondent hosts to dynamically discover the addresses of the HLRs serving a given mobile host. The first scheme introduces a set of Translation Server (TS) databases while the second uses a form of quorum consensus based on the Triangle Lattice (TL); for the latter we present an enhanced protocol called the Optimistic TL (OTL). For both schemes we present algorithms for mobile host registration and packet delivery, protocols for recovery from HLR failures, and complexity analysis of the overhead involved.

1. Introduction

There has been tremendous interest in the last few years in the areas of mobile and wireless communications. To provide these advanced services PCS and cellular systems, and the Public Switched Telephone Network (PSTN) in general, tend to use a "service node" architecture, where databases store the critical signaling information and intelligence, and switches are optimized for simplicity and high speed. In contrast, mobility and Quality of Service (QoS) support in the Internet are typically provided by means of enhancements or additions to the Internet Protocol (IP) [1] routers in the Internet fabric.

The Mobile IP protocol (MIP) [2] supports continuous Internet connectivity for mobile hosts (MH). An MH is always identified by the IP address it has when in its home network, called its home address. When a mobile host moves away from its home network it contacts a router called a Foreign Agent (FA) in the foreign network and obtains a temporary Care-Of-Address (COA). The COA may be the IP address of the FA, in which case it is called a co-located COA, or it may obtained from a separate entity, e.g. a Dynamic Host Configuration Protocol (DHCP) server [3]. The MH registers its COA with a Home Agent (HA), which is typically a router, in its home network, informing the latter of its COA. Any Correspondent Host (CH) wishing to communicate with the MH need not be aware the mobile host has moved; it simply sends IP packets addressed to the mobile host's home address. These packets are routed via normal IP routing to the mobile host's home network, where they are intercepted by the HA. The latter encapsulates each such packet in another IP packet which contains the mobile host's COA as the destination address, and these packets are thus delivered to the mobile host's new location (a process called *tunneling*.) Note that packets from the mobile host to the correspondent host need not necessarily be tunneled; the mobile host can simply send them directly to the correspondent host.

A well-known performance problem with MIP is that it uses "triangle routing", i.e., packets from the correspondent host to the mobile host must travel via three (sub)networks: the correspondent host's subnet, the home agent's subnet, and the subnet where the mobile host is currently located. An extension to the basic MIP protocol called Route Optimization (MIP-RO) [4] avoids triangle routing. However, packets sent by the correspondent host still use the triangle route until the correspondent host receives a binding update message from the HA with the mobile host's COA. In addition, the bandwidth and protocol overhead associated with tunneling of packets is still present. Another important implication is that, due to packet tunneling and triangle routing, interoperability problems arise when using protocols such as RSVP [5] for providing QoS guarantees to communications between correspondent and mobile hosts [6]. Similar problems arise with approaches like Diff-Serv [7].

*Address correspondence to: Ravi Jain, Bellcore, 445 South St., Morristown, NJ 07960. phone: (973) 829-3178. Fax: (973) 829-2645. Email: rjain@bellcore.com

In this paper we focus on the survivability aspects of MIP and MIP-RO, in particular their limitations when it comes to the fault-tolerance of a critical element of the protocols, namely the Home Agent. The first is the requirement that the Home Agent must reside in the home network of the mobile host. In MIP, after the mobile host registers, the Home Agent sends out a proxy ARP [1] to reply to ARP requests for the mobile host's IP address with it's (the Home Agent's) own link-layer address. A packet destined for the mobile host from anywhere in the Internet reaches the gateway on the mobile host's home network by usual IP routing, and then reaches the Home Agent because of the proxy ARP. If the Home Agent is not located in the home network normal IP routing will not deliver packets from the rest of the Internet to the Home Agent, but only to the gateway on the MH's home network.

Unless normal IP routing is modified or the usual intent and semantics of the Internet routing protocols are modified it seems difficult to overcome this limitation in MIP and MIP-RO. In situations where the home network is vulnerable to failure this becomes a serious problem. For example, consider mobile devices being used by military personnel in a battle zone. The home network of a particular user would be his or her company or platoon, but requiring the Home Agent to be in the home network would make the HA highly vulnerable. Instead, it is essential that it be possible to place the Home Agent at a secure distant location. Similar considerations arise for other scenarios (e.g. disaster relief). The second limitation of MIP and MIP-RO is that it is not possible at present to replicate the Home Agent at various locations distributed throughout the network in order to achieve survivability.

In sec. 2 we briefly describe our modification of Mobile IP, called MIP-LR, that uses databases called Location Registers (LR) for providing mobility support. MIP-LR avoids triangle routes and tunneling of data packets destined to the mobile host, and allows the mobility database to be placed outside the mobile's home network to improve survivability in situations where the home network is vulnerable. In sec. 3 we discuss how the survivability of MIP-LR can be enhanced further by using replicated mobility databases, and present two schemes for managing the replicated databases. Finally, in sec. 4 we end with some discussion and concluding remarks.

2. Mobile IP with Location Registers (MIP-LR)

We first describe MIP-LR assuming there is only a single HLR serving the mobile host. The major functions performed by the Home and Foreign Agents in MIP and MIP-RO are:

1. *Agent and network discovery*: Home and Foreign agents broadcast (or multicast) agent advertisements and respond to agent solicitation messages broadcast by the mobile host. This enables the mobile host to determine whether it is in its home or a foreign network.
2. *Database maintenance*: The Home Agent maintains the mapping from the mobile host's IP address to its COA. The Foreign Agent maintains the reverse mapping and/or the mapping from the mobile host's IP address to its link-layer (hardware) address in the foreign network.
3. *Tunneling*. The Home and Foreign agents encapsulate (and decapsulate) packets and forward them appropriately.
4. *COA allocation*. The Foreign Agent assigns a COA to the mobile host (if co-located COAs are not used.)

In MIP-LR we (1) eliminate the tunneling function; (2) use co-located COAs, so that the COA allocation function is performed by an external mechanism, like a DHCP server[1]; and (3) separate the remaining functions into different functional entities based upon the principle of separation of concerns. Thus we eliminate the Home and Foreign Agents, with their mix of functions. Instead, the database mapping the mobile host's IP address to its COA is maintained by a Location Register; by analogy with cellular systems [8,9], it is called a Home Location Register (HLR)[2]. MIP-LR is summarized as follows.

Subnet discovery. In MIP-LR there is a logical entity called the Advertisement Agent (AA), located in each subnet, which allows the mobile to discover which network it is in. The Advertisement Agent broadcasts (or multicasts) messages similar to MIP agent advertisements, and/or responds to solicitation messages broadcast by the mobile. (In practice, the Advertisement Agent may be co-located with the HLR where possible.)

Registration. The location of a mobile host is always registered at the HLR. When the mobile is at home the HLR simply maintains the identity mapping. When the mobile host moves to a foreign network it obtains a COA for that subnet; for concreteness in this description we assume that it obtains the COA from a DHCP server [3] for a time interval called a *lease*. The mobile host registers the COA with the HLR using a Registration message, as for MIP. The HLR returns a registration reply containing the allowed Lifetime for this registration (similar to MIP).

Packet Delivery. A Correspondent Host wishing to send a packet (see Figure 1) to the mobile host for the first time must first discover the IP address of the mobile host's HLR

[1] If a DHCP server or similar entity is not available, this function can be performed by a Visitor Location Registe (VLR); see [6] for details.

[2] Recall that in PCS systems an HLR contains parameters and features for mobile registration, authentication and service validation, vertical services (e.g. call forwarding and screening) and profile manipulation. See [9] for details.

(we will describe this process below, but for the moment note that it needs to be carried out infrequently.) The correspondent host then issues a query to the HLR, which returns the mobile host's COA as well as the remaining registration Lifetime (together called the mobility binding). The correspondent host then directly sends the packet to the mobile host's COA. The IP layer at the correspondent host thus hides the mapping from the mobile host's IP address to its COA from higher layer protocols (e.g. TCP [1]), and the IP layer at the mobile host does the same for the reverse mapping. The correspondent host caches the mobile host's binding in a binding Cache and uses the cached binding for subsequent packets destined to the mobile host. The CH must refresh its binding cache by querying the HLR again before the mobile host's remaining registration Lifetime expires.

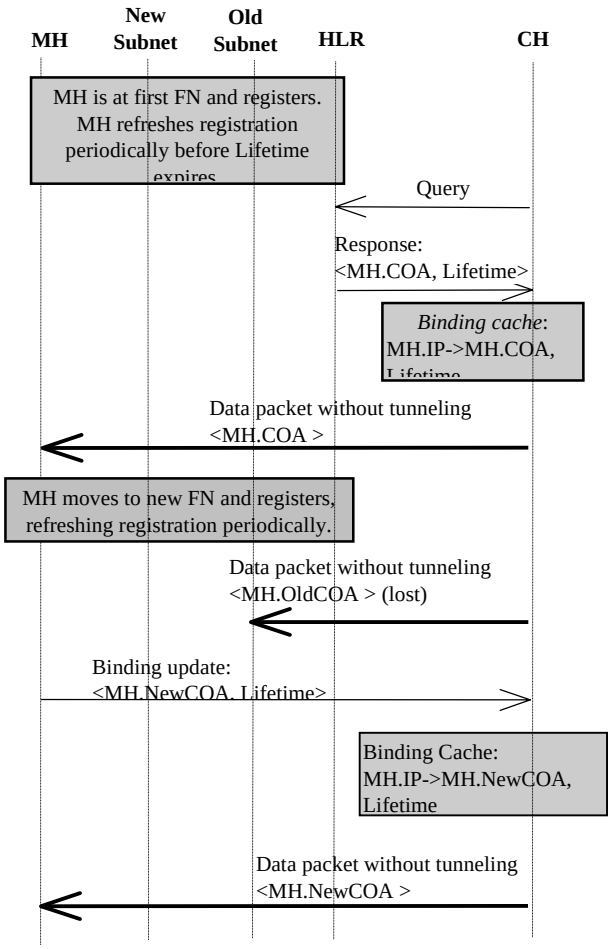

Figure 1: Packet delivery in MIP-LR with a single HLR. (Light arrows: control packets, heavy arrows: data.)

Cache maintenance. The mobile host maintains a list of *active* correspondent hosts, i.e., those that have sent messages to it during the current registration Lifetime. When the mobile host moves to a new subnet it informs these hosts of its new COA in the new subnet by means of a Binding Update message. This cache maintenance mechanism is similar to Mobile IP in IPv6 [10]. We refer to this as *eager caching* and omit further details here. Packets from the correspondent host which are in flight, i.e., after the mobile has moved but before the binding update has reached the correspondent, will reach the old subnet and be lost.

COA Deallocation. If the mobile host moved from one foreign subnet to another it may relinquish its COA in the old subnet by sending a message to the DHCP server which assigned the COA; if it fails to do so the old COA will be re-used after the lease expires in any case. While the mobile is located at a given subnet it must refresh its lease on its COA in that subnet by issuing request messages to the DHCP server before the lease expires, in accordance with standard DHCP procedure. The DHCP server will endeavor to assign the same COA for each such refresh request. However, if it assigns a new COA, the mobile host treats this as if it had just moved to the subnet, i.e., it registers with the HLR using the new COA and issues a binding update to its correspondent hosts.

Determining Mobility-Capable Hosts. In MIP and MIP-RO the correspondent host does not take any special action when communicating with mobile hosts. MIP-LR sacrifices this transparency for improved performance, survivability and interoperability with protocols for QoS. Thus in MIP-LR the correspondent host must issue a query to the HLR to discover the mobile's current address. This may be quite reasonable for environments like military tactical applications. Otherwise, the CH must determine which hosts are mobility-capable. Different possible approaches to doing this are to (1) reserve portions of the address space for mobility-capable hosts, and (2) to use an off-line discovery mechanism, e.g. a notification or directory scheme. We have briefly discussed these previously [6] and will omit further discussion here due to lack of space.

Obtaining the HLR address. How the correspondent host obtains the address of the mobile host's HLR is related to the issues of HLR survivability. If the HLR serving the mobile host was located in the mobile's network, the correspondent host could issue the query for the COA to the mobile host's permanent IP address; if the mobile host were at home, it would respond, and if not, the HLR would trap the query (using proxy ARP) and respond.

However, for survivability the HLR may not be located in the mobile's home network. In that case, two possibilities arise: that a single HLR suffices, or multiple HLRs are required. In a small system it might be possible that a single

HLR suffices to support all the mobile hosts in the system. In that case every correspondent host may be configured with the address of the HLR; to improve survivability, the HLR can be implemented on a commercial hot-standby fault-tolerant platform. In general, however, multiple HLRs may be required to distribute the workload, to reduce the network communication cost of contacting the HLR, and to obtain survivability, by scattering the HLRs throughout the network. In section 3 we suggest two possible approaches for maintaining the multiple HLRs.

2.1 Discussion of MIP-LR Design

Elimination of Foreign Agents. Unlike MIP and MIP-RO, in MIP-LR there are no FAs. In addition to the arguments for eliminating Foreign Agents and using co-located COAs discussed in previous schemes such as MosquitoNet [11], there are some specific to MIP-LR. Firstly, since there is no tunneling, no decapsulation is required. Secondly, we believe allocation of COA is best separated out and handled by existing methods to perform dynamic address allocation, e.g. DHCP. And for subnet discovery we place a simple Advertisement Agent (AA) in each subnet.

Eliminating the FA helps improve survivability; although an Advertisement Agent is still introduced specifically to support mobility, note that its functions (issuing broadcasts specifying the identity of its subnet and responding to agent solicitation messages) are much simpler than those of a FA. This makes failure recovery simpler as virtually any surviving host can take over as an AA. As in MIP, if a mobile host does not receive agent advertisement messages for some period of time, it broadcasts an agent solicitation message to the entire subnet. Each host acting as a backup AA waits for a random interval of time, and if it does not hear any agent advertisements, begins broadcasting agent advertisements. If two Advertising Agents start broadcasting simultaneously, binary exponential backoff is used to resolve the conflict (or some other scheme can be used, e.g. the AA with the higher IP address wins.) Situations where multiple Agents continue to broadcast (e.g. due to hidden terminal problems) can be resolved, although we omit discussion for brevity.

Lazy caching. The caching scheme we have employed requires the mobile to update the correspondent's cache when it moves. An alternative is lazy caching, such as that employed in MIP-RO where, after a mobile moves, the Foreign Agent in the old subnet traps packets destined for the mobile hosts, and (optionally) tunnels them to the new subnet; this is also used as optimization for handling packets in flight, i.e., those sent by the correspondent after the mobile has moved. However, this protocol requires an entity in the old subnet; we have chosen to forego this optimization in return for improved survivability.

Interoperability with MIP. MIP-LR is designed for closed enterprise environments, but can interoperate with hosts outside the environment using MIP as follows. Packets from correspondent hosts outside the system are trapped at the border gateways, which query the HLR to obtain the mobile's COA and tunnel the packet to the mobile; the mobile must have the ability to decapsulate packets. This method of interoperability is similar to what is carried out by Network Address Translators (NAT). MIP-LR can also interoperate with MIP hosts inside the closed enterprise network provided a Home Agent is installed in the mobile's home network; we omit details for brevity.

3. Managing Multiple HLRs

Multiple HLRs may be introduced for survivability reasons (and possibly to improve performance and reduce network resource consumption.) We present two alternative methods for maintaining multiple HLRs.

3.1 Translation Servers

We introduce databases, called *Translation Servers* (TS), which store the mapping from a host's IP address to the IP address of the HLR serving that host. Since this information does not change frequently, a correspondent host can cache the response for relatively long periods of time.

In the context of this approach consider the situation where there are layers of vulnerability in the network, e.g. a military scenario where host machines belong to various levels of the military hierarchy (brigade, platoon, squad, etc.) There are more hosts at the lower levels of the hierarchy than at higher levels; typically they are more mobile and more vulnerable. It is desirable to place the HLR outside the home network for hosts at a low level in the hierarchy; however, this increases the cost of mobile host registration (and, in general, of packet delivery to mobile hosts). We thus assume that HLRs are placed at higher-level locations that are less vulnerable than those of individual hosts, and TS are placed at higher levels still, which can be regarded as being essentially secure. (Again, for fault-tolerance, the TS may be implemented on hot-standby platforms.)

For a variety of reasons (recovery from HLR failure, load balancing, or to reduce the costs of communicating with the HLR) it will be desirable to be able to change the HLR serving a given mobile host. Maintaining the mapping from a mobile host's IP address to its HLR's IP address in a TS provides a convenient mechanism for dynamically reassigning mobile hosts to HLRs. Notice that the mobile host itself need not know its own HLR's address; the mobile can discover it dynamically by querying the TS also.

3.1.1 Single Translation Server

For the moment, assume that a single TS suffices for the entire system, and its address is known to all hosts. Then the mobile host performs registration using the procedure given in pseudo-code below, where $r > 0$ is the number of HLRs the mobile host must register at for survivability. Note that the same procedure is used when the mobile registers because it has moved or because its current registration Lifetime is about to expire. Registration requests and responses use datagrams to avoid the overhead of connection setup but are retransmitted when timeouts occur to provide a degree of reliability.

```
register(r){
  Registration_state = TRYING;
    /* MH.IP: MH's permanent IP address. */
    /* HLR_list: MH's current HLRs (known).
*/
    /* r: total number of HLRs desired */
  if(length(HLR_list) < r)
    HLR_list = query_TS(MH.IP, HLR_list, r);
  Pending_HLR = HLR_List;
  while(Registration_state != SUCCESS){
    Retries = 0;
    while(Retries++ < MAX_RETRY){
       for(all HLRs in Pending_HLR)
         Issue timestamped registration
request
       if(ACK received from all Pending_HLRs
              before timeout){
         Registration_state = SUCCESS;
         return;
       }
       else if(ACK received from some HLRs
              before timeout){
         Registration_state =
PARTIAL_SUCCESS;
         Update Pending_HLR list
       }
    }
       /* One or more HLRs has failed */
       /* TS returns other HLRs and */
       /* updates its own records */
    Pending_HLR
       = query_TS(MH.IP,
              HLR_List – Pending_HLR, r);
    Update HLR_List
  }
}
```

In the *register* procedure the choice of the number of HLRs at which the mobile must register (i.e., *r*) can be left to the mobile, or it can be left to the TS with the mobile providing a hint in its query request (i.e., *query_TS()*) based upon its current requirements. The choice of which particular HLRs serve the mobile host is left to the TS. This is because we assume that the TS will have better knowledge of the availability of HLRs in the system than the mobile, and may assign mobile hosts to HLRs based on a variety of criteria, e.g. proximity, load balancing, etc. (The design of this assignment algorithm is outside the scope of this paper.) Once at least one registration succeeds (i.e., state PARTIAL_SUCCESS is reached) the mobile can allow other activities (e.g., communicating with other CH) to continue in parallel.

```
find(MH.IP, r){
  if(unexpired COA exists in Binding Cache)
    return(COA);
  Pending_HLR = null;
  if(HLR_List Cache does not have MH's
       HLR_List)
    HLR_List
       = query_TS(MH.IP, Pending_HLR, r);
  Phase = 0;
  while(Phase++ < MAX_PHASE){
       /* Choose some HLR(s) to query */
    Pending_HLR = select(HLR_list);
    Retries = 0;
    while(Retries++ < MAX_RETRY){
       Issue binding request to all
Pending_HLR
       if(reply received from any pending
HLRs
              before timeout){
         Put COA & Lifetime for the reply
with
              latest timestamp in Binding
Cache
         Update HLR_List in HLR_List Cache if
              necessary
         return(COA);
       }
       /* Pending HLRs have all failed. */
       /* Retry by some policy, e.g. Binary
          Exponential Expansion */
       Pending_HLR
         = retrypolicy(Phase, Retries, r);
    }
       /* All HLR(s) in HLR_List have
failed.*/
       /* Query TS for m alternates */
    m = retrypolicy(Phase, Retries, r);
    HLR_list
       = query_TS(MH.IP, Pending_HLR, m);
    Pending_HLR = null;
  }
  return(FAIL);
}
```

A correspondent host wishing to contact the mobile host (for the first time or subsequent times) carries out the *find* procedure. The correspondent obtains a list of *r* HLRs serving the mobile from the TS. (The number as well as the identity of HLRs which the TS actually returns could be left up to the TS.) This list is maintained in a cache, called the

HLR_List cache, separate from the Binding Cache which keeps the mobile's COA. The correspondent selects and queries a subset of the HLRs assigned to the mobile, and uses the COA with the latest timestamp. If a choice is available the correspondent selects an HLR to query using any criterion it chooses (proximity, randomly, etc.) Since the mobile host always registers at all the HLRs it is assigned to, the strategy where the correspondent host chooses only one HLR out of the HLRs returned by the TS is called *Write-all-Read-any*.

Binary Exponential Expansion. The number of HLRs requested by the correspondent, as well as the strategy it uses to query the HLRs, reflects a balance between latency and resource consumption. If latency is to be minimized, the correspondent requests all the HLRs serving the mobile and queries all of them in parallel; to minimize resource consumption, the correspondent queries one HLR at a time. We suggest the following *Binary Exponential Expansion* policy as a compromise: the correspondent host requests the addresses of all HLRs ($r = ALL$), and at each retry (*Retries* is incremented), it queries $2^{Retries}$ HLRs in parallel.

Race conditions exist in the protocol when failures occur, so that for short intervals the HLRs assigned to a mobile may contain inconsistent information and the correspondent may obtain incorrect results (similar situations occur in MIP and MIP-RO). Our approach relies upon retransmissions at the CH instead of complicating the protocols.

A mobile host contacts a TS only: (1) the first time it is installed, if it is not pre-configured with its list of HLRs; (2) if HLR failures occur and replacements are required. A correspondent host contacts a TS only: (1) when sending a packet to a mobile for the very first time; (2) if the mobility binding for a host it wishes to contact has expired or been deleted from the cache *and* the list of HLRs serving the mobile has expired or been deleted from the *HLR_List* cache; (3) if HLR failures occur and alternate HLRs are required.

To summarize, the scheme in this section: (1) introduces a TS with an address known by all hosts in the system; (2) for the HLRs assigned to a given mobile host, a Write-all-Read-any protocol is used; (3) in case of HLR failure, the mobile requests alternative HLRs from the TS, and other activities at the mobile can continue; (3) in case of HLR failure, the correspondent uses a Binary Exponential Expansion policy.

3.1.2 Multiple Translation Servers

For most closed network scenarios a single (fault-tolerant) TS probably suffices. For additional survivability multiple TS may be deployed using the following scheme. (This scheme is based on a method we have presented in [12].)

Dynamic Hashing Scheme. Each host in the system knows the addresses of all the t TS in the system, and in addition has a hash function f which maps a mobile host's IP address to one or more integers j, where $0 < j < (t + 1)$, and $TSaddr(j)$ is the IP address of a TS serving the mobile. When the set of HLRs serving a mobile host is to be updated (for load balancing, failure recovery, addition of new HLRs, etc.) then all the TS serving the mobile are updated. However, to find the set of HLRs serving the user, only one of the TSs need be queried; the scheme thus favors TS queries over updates.

Changing the set of set of HLRs serving a mobile does not require changing any information (the TS addresses or the hash function f) at any host. This information is only modified if the set of TSs is changed (an infrequent operation) or more TSs are added (an even more infrequent operation). Even in that case, modifying the hash function itself can be avoided, using a technique based on dynamic hashing [13] as follows. (For simplicity we assume each mobile is served by a single TS; extension to service by multiple TS is straightforward.) Let f return a large number of bits, and we use k of them (e.g. the least significant k bits) as an index into *TSaddr*, the table of TS addresses. Now suppose a particular TS, $TS(j)$, is overloaded and a new TS is to be added. At $TS(j)$, the hash function f is applied to the address of each mobile served by that TS, except now instead of using k bits returned from f, we use $(k + 1)$ bits. The extra bit returned by f is examined; if it is zero the record for that mobile (mapping the mobile's IP address to the list of its serving HLRs) is moved to the new TS; otherwise, it remains in the old TS. Once all the records in $TS(j)$ have been thus processed, the address of the new TS is broadcast to all hosts and each host increments k; the new TS can now offload $TS(j)$.

Observe that this process does not require modifying the HLRs, or any of the TSs that is not overloaded, or any modification to the software in any host, making it highly suitable for on-line system upgrade. Its additional processing cost is $O(m/t)$ at the TS, where m is the number of mobiles and t is the number of TS, $O(1)$ at all the hosts in the system, and it requires a single system-wide broadcast message.

3.2 Quorum Consensus (QC)

An alternative to Translation Servers is Quorum Consensus (QC), previously proposed for maintaining replicas in distributed database systems [14]. For our application we assume that every host in the system knows the address of all HLRs in the system. We first describe Basic QC.

The Basic QC Algorithm. Basic QC defines a Read Threshold (RT) value and a Write Threshold (WT) value such that $h < RT + WT < (2h + 1)$ and $h/2 < WT < (h + 1)$,

where *h* is the total number of HLRs. A mobile host must ensure that it registers its COA, along with a timestamp, with *any* WT number of HLRs; a correspondent host must read *any* RT number of HLRs and use the value with the latest timestamp. The conditions on WT and RT ensure that the set of HLRs written by the mobile host (the *write quorum*) always intersects with the set of HLRs read by the correspondent host (the *read quorum.*) The mobile and correspondent hosts can use any suitable criterion (randomization, proximity, etc.) to select which HLRs belong to the read and write quorum. The QC algorithm can be implemented within the general framework of the routines *register()* and *find()*in Section 3.1.1, since it basically replaces the *query_TS()* routine. The Binding Cache and the *HLR_List* cache are still maintained. We omit further details.

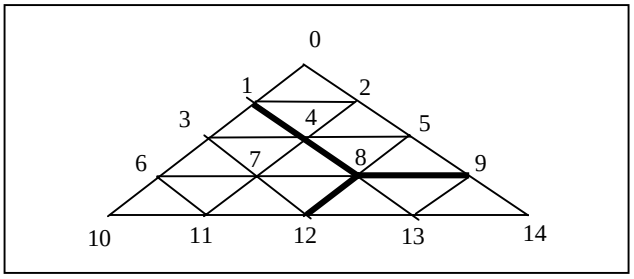

Figure 2: Triangle Lattice (TL) system for *d = 5*

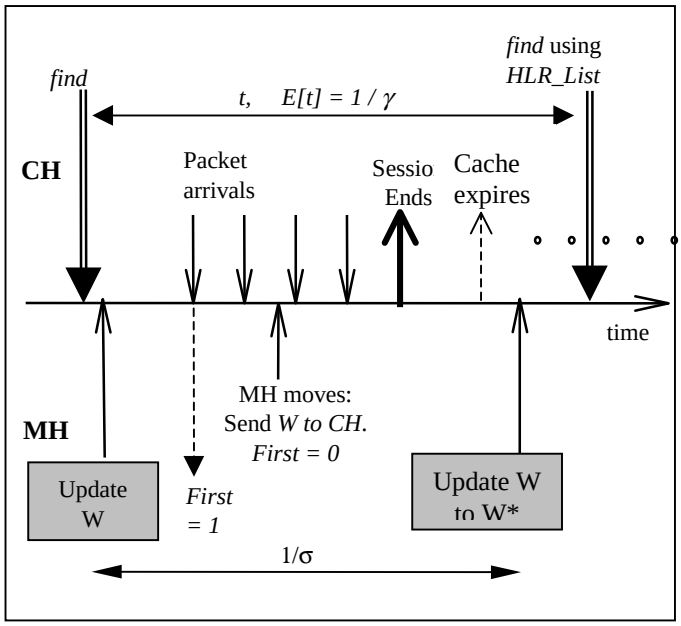

Figure 3: Timing diagram illustrating Optimistic TL

Quorum algorithms are evaluated using several criteria [15]. The *quorum size*, defined as the size of the smallest quorum in the system, determines the cost of accessing the members of the quorum. The *cost of failures* is the additional number of processors that must be contacted in order to establish a quorum when failures occur. For Basic QC algorithm, the quorum size and the worst-case cost of a failure is *O(h)*. We use a *planar quorum* algorithm, specifically a Triangular Lattice TL [16] and develop an improved version called Optimistic TL (OTL).

In TL (see Figure 2) the processors (in our case, HLRs) are assumed to be at the vertices of a virtual triangle lattice. A quorum consists of a minimal path connecting two of the sides of the triangle, and a second minimal path connecting the first minimal path to the third side, e.g., in Figure 2 the set {1, 4, 8, 9, 12} is a quorum. It is easy to see that any two quorums intersect and that quorum size for TL equals *d*, the length of a side of the triangle, i.e., if there are *h* HLRs, $h = (d^2 + d)/2$, so the quorum size $d \approx \sqrt{(2h)}$. The cost of a failure is *O(1);* in fact it is 6, since at most 6 processors have to be contacted to find an alternate processor.

3.2.1 Optimistic TL (OTL) algorithm

With most quorum systems, including TL, a relatively large number of processors must be contacted even when there are no failures. For our application, in the case of Basic QC *O(h)* HLRs must be contacted; TL reduces this to $O(\sqrt{h})$. For registrations this is acceptable since the timing and extent of failures is unpredictable; for queries, however, the number of HLRs contacted should be reduced without increasing the latency of obtaining the mobile's COA. Unlike general quorum applications (mutual exclusion, replica control, etc.), where the quorum reader and writer generally do not communicate directly, in our application the reverse is true. We exploit the special nature of our application to develop Optimistic TL (OTL) which consists of two simple optimizations to TL (Fig. 3.):

1. Mobile informs correspondents of quorum. In MIP-LR (as in Mobile IP for IPv6 [10]) the mobile maintains a list, *Active_CH*, of correspondent hosts with which it is in active communication within the current registration Lifetime. The mobile adds to each entry in this list a single bit, *First*, which is set when the correspondent host is first added to the list. When the mobile changes registration areas it sends binding updates to all correspondent hosts on the *Active_CH* list as usual. In addition, for hosts with *First* set, the mobile includes in the binding update message the indices of the HLRs in its current quorum, thus informing the correspondent of its entire quorum, and resets *First* (similarly if the mobile refreshes its registration before it has moved.)

2. Correspondents cache mobile's quorum. The first time a correspondent host searches for a mobile, it must contact all the HLRs in the quorum. However, as a result of this query the correspondent becomes aware of the set of

HLRs, S, $|S| > 0$, that lies at the intersection of the read and write quorums, namely, the HLR(s) with the latest value of the mobile's COA. For subsequent queries, the correspondent contacts only one of the intersecting HLRs, say H in S. Further, the first time the mobile moves while it is in active communication with the correspondent, it informs the latter of its entire quorum, and the correspondent adds these to S. If there is no response within the timeout period when the CH queries H, it queries one of the HLRs in $|S - H|$ if $|S - H| > 0$, or the 6 HLRs adjacent to the failed node otherwise.

Complexity of OTL. The additional storage complexity of OTL over TL is $O(c)$ bits at the MH, where c is the number of active CH, and $O(m\sqrt{h}\log h)$ bits at the CH, where m is the size of the *HLR_List* cache, i.e., the number of mobiles for which the correspondent maintains a list of HLRs. The increase in the size of the binding update message sent by the mobile (or a new message) is $O(\sqrt{h}\log h)$ bits.

In exchange, the correspondent generally contacts only a single HLR, and the CH can reduce the average cost of failures considerably. Let C_{HLRQ} be the cost of querying an HLR. Then the expected cost of a single HLR failure with TL is $F_{TL} = 6\,C_{HLRQ}$. The corresponding value F_{OTL} for OTL can be derived under certain assumptions as follows. Suppose the correspondent host initiating a session to a given mobile host is a Poisson process with mean $1/\gamma$ seconds, $1/\gamma >> Lifetime$, where *Lifetime* is the average registration lifetime of the mobile host. Assume the CH maintains an entry in the *HLR_List* cache for a time much longer than $1/\gamma$. Let the time between the mobile's write quorum changes (due to HLR failures, etc.) be exponentially distributed with mean $1/\sigma$. Then it can be shown [17] that $F_{OTL} = (1 + \sigma/(\gamma+\sigma))\,C_{HLRQ}$. We expect $\gamma >> \sigma$, but even if $\gamma = \sigma$, OTL reduces the expected cost of a single failure by a factor of 4 over TL.

QC incurs more processing, storage and message complexity costs than TS, in general, but reduces the latency experienced by the CH for obtaining the mobile's COA as well as the expected message complexity and latency of a single HLR failure. It also involves less management overhead.

4. Related work and discussion

There has been substantial previous work on mobility management schemes for PCS as well as IP networks. For locating mobile users in ATM networks, a scheme somewhat similar to MIP-LR, called LR, has been developed [18], but is specifically relevant only to the architecture of PNNI.

We have previously [6] presented a scheme that introduces Location Registers and presented an analysis of the scheme that indicates it can result in significant reductions in the mean total network costs compared to MIP. However, the scheme in [6] does not allow HLRs to be replicated or placed outside the home network, and has no provision for survivability. It also assumes the existence of VLRs (roughly analogous to MIP's Foreign Agents) in the visited subnet.

In the present paper we present the design of a protocol, MIP-LR, that provides improved performance over MIP by avoiding triangle routing and encapsulation of data packets, interoperability with protocols for QoS, and enhanced survivability. MIP-LR provides better survivability by: (1) Eliminating Foreign Agents and replacing them with a simple Advertisement Agent and recovery protocol; (2) Allowing HLRs to be placed outside a (vulnerable) home network, and (3) Allowing HLRs to be replicated.

We have presented two schemes for maintaining replicated HLRs and for obtaining the address of an HLR serving the mobile host: one which introduces Translation Server (TS) databases and uses dynamic hashing, and one based on Quorum Consensus (QC), specifically one where the HLRs are organized into a virtual Triangular Lattice (TL); for the latter we have presented an enhancement called Optimistic TL (OTL). The choice of algorithm is determined by several factors that must be evaluated for each deployment scenario.

MIP-LR operates well within enterprises or within logical administrative domains (e.g. tactical military environments). However, it can interoperate with MIP and MIP-RO also. Like MIP-RO, MIP-LR requires correspondent hosts to be aware of host mobility; however, unlike MIP-RO, it allows interoperability with RSVP and avoids packet encapsulation. MIP-LR essentially separates out the database functionality of Home Agents, providing more flexibility for offering advanced services. Finally note that the TS and QC schemes are applicable to PCS and cellular systems also.

Acknowledgments. We thank Tony McAuley, Archan Misra, and Sue Thomson of Bellcore for their comments and many useful discussions. We also thank the reviewers for their comments.

References

[1] Stevens, W. R. *TCP/IP Illustrated*, Addison-Wesley, 1994.

[2] Perkins, C., (ed.), "IP Mobility Support", *RfC 2002*, Oct. 1996.

[3] Droms, R., "Dynamic Host Configuration Protocol", *Internet RfC 1541*, Oct. 1993.

[4] Johnson, D. B., C. Perkins (eds.), "Route Optimization in Mobile IP", *Internet Draft*, draft-ietf-mobileip-optim-06.txt, Nov. 1997.

[5] Braden, R., et al, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," *Internet draft*, draft-ietf-rsvp-spec-16.txt, June 1997.

[6] Jain, R., T. Raleigh, C. Graff and M. Bereschinsky, "Mobile Internet Access and QoS Guarantees using Mobile IP and RSVP with Location Registers," *Proc. ICC '98*, June 1998.

[7] Blake, S., et al, A Framework for Differentiated Services, *Internet Draft*, draft-ietf-diffserv-framework-01.txt, Oct. 1998.

[8] Jain, R. Y.-B. Lin and S. Mohan, "Location strategies for Personal Communications Services". In J. Gibson (ed.), *Mobile Communications Handbook*, CRC Press, 1996.

[9] Bellcore, PCS Network Access Services, *SR-TSV-002459*, 1994.

[10] Perkins, C., and D. Johnson, "Mobility support in IPv6," *Proc. Mobicom '96*, 27-37, Nov. 1996.

[11] Baker, M., X. Zhao, S. Cheshire and J. Stone, "Supporting mobility in MosquitoNet," *Proc.USENIX '96*, Jan. 1995.

[12] Jain, R., S. Rajagoplan, and L. F. Chang, "Phone number portability for PCS systems with ATM backbones using distributed dynamic hashing," *IEEE J. Sel. Areas Comm.*, 96-105, Jan. 1997.

[13] Fagin, R., J. Nievergelt, N. Pippenger and H. R. Strong, "Extendible hashing – A fast access method for dynamic files," *ACM Trans. Database Sys. 4, 3, 315-344, Sep. 1979.*

[14] Bernstein, P., V. Hadzilacos and N. Goodman, *Concurrency Control and Recovery in Database Systems*, Addison-Wesley, 1987.

[15] Naor, M. and A. Wool, "The load, capacity and availability of quorum systems," *Proc. IEEE Symp. Foundations of Computer Science*, 214-225, 1994.

[16] Bazzi, R., "Planar Quorums," *Proc. Tenth Intl. Workshop on Distributed Alg.*, Springer-Verlag, Oct. 1996.

[17] Nelson, R. *Probability, Stochastic Processes and Queuing Theory*, Springer-Verlag, 1995.

[18] Dommety, G. and M. Veeraraghavan, "Location management in wireless ATM networks", *Proc. WINLAB Workshop*, Mar. 1997.