Simultaneous Mobility: Analytical Framework, Theorems and Solutions

K. Daniel Wong,* A. Dutta,† H. Schulzrinne‡ and K. Young

July 18, 2005

Abstract

The original Mobile IP (MIP) protocol does not perform route optimisation but uses Home Agents to forward traffic. Thus, it does not have problems with simultaneous mobility, i.e., the special case when both end hosts are mobile and move at about the same time. However, MIP for IPv6 (MIPv6) uses binding updates that are sent directly to a correspondent node. Session Initiation Protocol based mobility management (SIPMM) and MIP with Location Registers (MIP-LR) also use direct binding updates between a Mobile Host and a correspondent node. Thus, MIPv6, MIP-LR, and SIPMM are vulnerable to the simultaneous mobility problem. In this paper, we analyse the simultaneous mobility problem and solution mechanisms, and propose new ways for MIPv6, MIP-LR and SIPMM to handle simultaneous mobility.

1   Introduction

Many schemes for mobility management of IP nodes have been proposed. Seven "properties to fully realize the promise of ubiquitous mobility" are proposed in [1], including *simultaneous mobility*, i.e., that "end hosts should be able to move simultaneously without breaking an ongoing session between them." [1]

MIP-LR [2, 3] and Session Initiation Protocol (SIP [4]) related solutions have advantages over the original[1] Mobile IP (MIP [5]) scheme. For example, they avoid a single-point-of-failure by allowing replication of location information in multiple Home Location Registers (HLRs, used in MIP-LR) and SIP servers. They solve the triangular routing problem by sending binding updates directly to correspondent nodes and receiving packets directly from correspondent nodes without going through another node like a Home Agent. However, SIP Mobility Management (SIPMM) and MIP-LR also have their weaknesses. For example, in some cases they may not handle mobility management for real-time and non-real-time traffic equally well (but a possible solution is to use both of them together [6]). In this paper, though, we focus on a particular shortcoming of SIPMM and MIP-LR: they currently do not handle simultaneous mobility well.

Meanwhile, the triangular routing problem of the original MIP has become the quadrilateral routing problem of MIPv6 [7], as bi-directional tunnelling both to and from the Home Agent replaces the unidirectional tunnelling from the Home Agent of the original MIP. However, this drawback is offset by the increased emphasis on route optimisation. Whereas route optimisation was not part of the original MIP but just an Internet draft [8], it has become an integral part of MIPv6. However, when route optimisation is used in MIPv6, MIPv6 becomes vulnerable to the simultaneous mobility problem.

---

*Department of Information Technology, Malaysia University of Science and Technology, GL-33 Block C Kelana Square, 47301 Kelana Jaya, Petaling Jaya, Selangor, Malaysia; email: dwong@must.edu.my

†Dutta and Young are with Telcordia; email: {adutta,kcy}@research.telcordia.com

‡Columbia University; email: hgs@columbia.edu

[1]It is original in the sense that there is now a version for IPv6, that we abbreviate as MIPv6.

In this paper, therefore, we analyse the simultaneous mobility problem for SIPMM, MIP-LR and MIPv6, and propose solutions. This is a serious problem for the following reasons:

1. These schemes, especially MIPv6 and SIPMM, are expected to be widely deployed in the next generation IP networks for mobility management

2. The probability of simultaneous mobility can be surprisingly high in certain cases that could realistically occur, e.g., with high handoff rates and long handoff latency as will be seen in Section 3.3

To be precise, when we speak of movement of an end host, we refer to the kind of movement that is not transparent to the network layer and necessitates a network layer handoff. Thus, movement that is transparent to the network layer, including "link layer handoffs" (e.g., movement within a WLAN extended service set), is not of interest in this paper. Simultaneous mobility refers to two communicating end hosts moving (both of them in such a way as just described) at about the same time. We will be more precise in Section 2. Non-simultaneous mobility refers to mobility of one end host while the other remains stationary.

We would expect that non-simultaneous mobility would in most scenarios occur more frequently than simultaneous mobility[2]. Nevertheless, it would happen once in a while and must be handled properly by mobility protocols. We define the simultaneous mobility problem to be the problem of losing a binding update from one mobile node because it is sent to a previous address of the other mobile node that is also moving at around the same time. The disruption caused by the simultaneous mobility problem may far exceed the typical disruption caused by non-simultaneous mobility. Older protocols like the original MIP handle simultaneous mobility adequately, because of non-mobile home agents. Therefore, in this paper, we analyse the simultaneous mobility problem for MIPv6, SIPMM and MIP-LR, and propose solutions using a common approach, noting that the problem simultaneous mobility is very similar in these protocols. Our solutions are designed to impose minimal changes on the existing protocols while efficiently dealing with the simultaneous mobility problems.

We focus on situations where the handoff rate of a mobile node is typical enough that consecutive handoffs of the same mobile node are non-overlapping. We do not focus on situations of overlapping consecutive handoffs of the same mobile node, where one handoff has not completely finished before the next begins, e.g. there has not been enough time after the acquisition of an IP address for binding updates to reach their destination networks.

The reasons for our focus are as follow:

1. The problems encountered with overlapping consecutive handoffs are not so much a problem of simultaneous mobility as a problem with excessive handoff rate. Whether the correspondent node is mobile or fixed, there will be very severe problems when the mobile nodes changes its IP address before binding updates for its previous IP address have even arrived at their destinations.

2. For the foreseeable future, the extreme case of handoff rates high enough for overlapping consecutive handoffs is highly improbable.

Hence, consecutive handoffs of a mobile node are non-overlapping in this paper, and we focus rather on overlap of handoffs of different mobile nodes, i.e. simultaneous mobility.

Reference [9] extends a TCP migration mobility protocol [10] to handle simultaneous mobility, but there are significant differences between the TCP migration schemes (where mobility is handled at the transport layer) and MIP-related protocols or SIPMM. A scheme for handling simultaneous mobility (and other issues) at a layer between the transport and application layers is presented in [11]. In that scheme, mobility is handled mainly at the transport layer, using Stream Control Transmission Protocol (SCTP) extensions. We have presented initial thoughts on possible solutions to the simultaneous mobility problem for MIP-LR, SIP and MIPv6 in [12, 13]. However, no analytical framework or theorems and proofs related to the simultaneous mobility problem for SIP, MIPv6 and MIP-LR have been proposed before. We

---

[2]An uncommon scenario where simultaneous mobility might outnumber non-simultaneous mobility is where both end hosts move very frequently so the likelihood of simultaneous mobility is high. Another is when there is some degree of synchronisation between mobility on both sides.
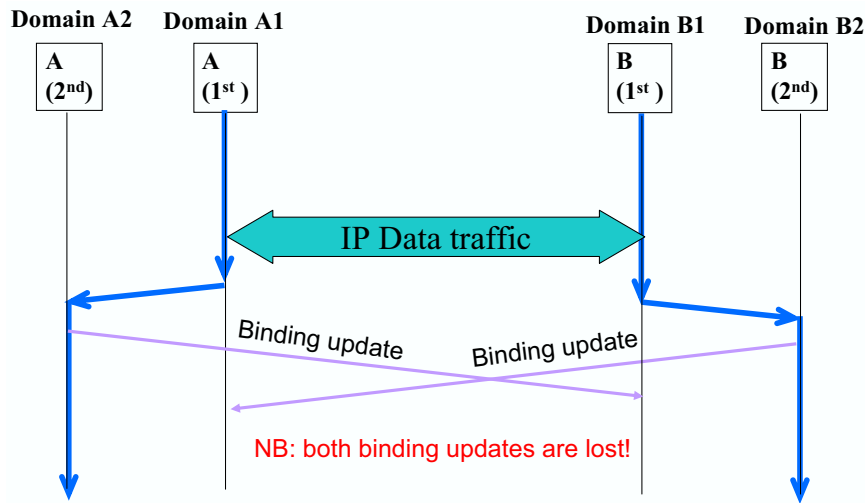
Figure 1: The Simplest Case of the Simultaneous Mobility Problem Illustrated

illustrate the simultaneous mobility problem in Section 1.1. We follow this somewhat informal illustration of example scenarios with a more formal treatment of the problem in Section 3 after proposing an analytical framework in Section 2. We analyse the solution mechanisms in Section 4, then propose solutions in Section 5.

1.1 Illustration of problem

Note that in this paper the kind of mobility of interest is terminal mobility (rather than other notions of mobility like personal mobility, service mobility, etc.) of end hosts (it may be assumed that router mobility would be handled by other means, e.g. ad hoc routing protocols in conjunction with auto-configuration protocols). Both pre-session and mid-session mobility are considered, although we concentrate more on mid-session mobility. We focus on layer 3 handoffs, i.e. where IP address changes are involved. The basic simultaneous mobility problem is shown in Figure 1. There are two nodes, A and B. Time is in the vertical direction (and flows downward), whereas spatial location is in the horizontal direction. A moves from domain A1 to A2 while B moves from domain B1 to domain B2. After their respective moves, they send binding updates to the other node, and both binding updates are lost. No proxies (e.g., location proxies, to be defined later) are involved, so it is the simplest, basic case of the simultaneous mobility problem occurring.

For layer 3 terminal mobility of end hosts, the original MIP was the earliest wide-spread solution. However, it has weaknesses such as the Home Agents being single points of failure. MIP-LR, on the other hand, uses replicated databases (Home Location Registers) that replace Home Agents as agents and are not single points of failure. However, the Home Location Registers are only queried to set up a communications session. Subsequently in the communications session, changes in location are communicated directly between the two mobile nodes. SIP, meanwhile, is designed to manage real-time sessions, e.g. packet-switched voice and video sessions. Mobility of the end hosts is handled by sending re-INVITE messages to the other party, informing it of its new location (e.g., its new IP address). Thus, in both MIP-LR and SIP, direct binding updates are sent between the mobile nodes.

Figure 1 is easily adapted to illustrate the simultaneous mobility problem with SIP, as seen in Figure 2. The binding updates become re-INVITE messages. The main difference is that two SIP servers are added, one for each mobile node in its home network. Whenever a mobile node moves, it sends a registration message to its home network SIP server. This server can thus serve as an up-to-date location proxy (location proxies will be defined shortly in Section 2.4) for the mobile node. However, in general, these servers are not involved in preventing the simultaneous mobility problem. The re-INVITE messages are lost and
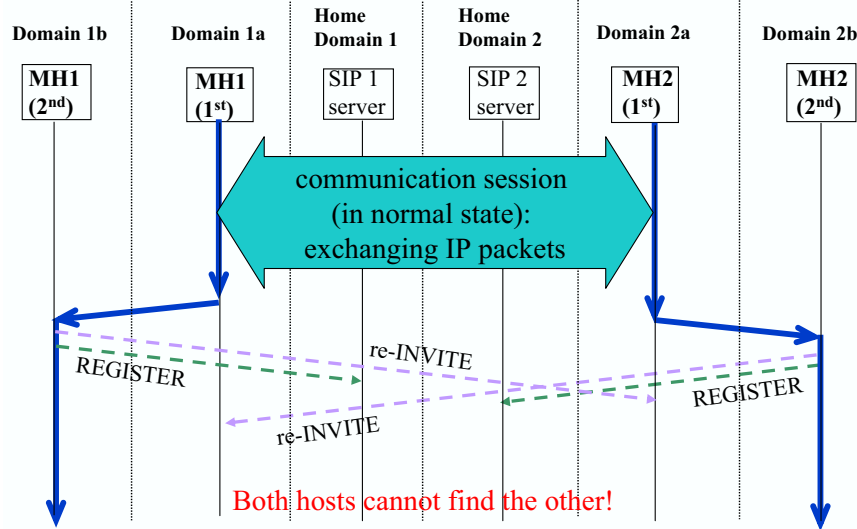
Figure 2: Simultaneous Mobility Problem for SIP Illustrated (and for MIP-LR too, with small modifications)
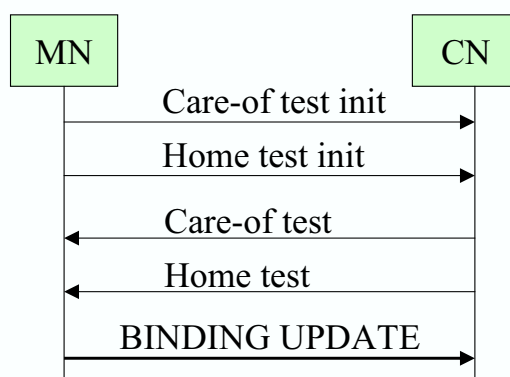

Figure 3: Return Routability in MIPv6

the servers do not intervene. Only new potential correspondent nodes hoping to locate the mobile node benefit (the reader may wonder whether the involvement of the SIP servers in the re-INVITE signalling exchange could be used in solving the simultaneous mobility problem – indeed, this is a feature of some of the solution mechanisms that will be discussed in Section 4). Furthermore, Figure 2 could also be used to illustrate MIP-LR's simultaneous mobility problem. We just need to replace the SIP servers with MIP-LR home location registers, and re-INVITE with MIP-LR binding updates.

Meanwhile, concerns about non-optimal routes in MIP led to the introduction of route optimisation (MIP-RO). MIP-RO is not vulnerable to the simultaneous mobility problem, as we will prove in Section 3. On the other hand, MIPv6 is vulnerable to the simultaneous mobility problem, as we will also prove in Section 3. The direct binding updates from the mobile node to the correspondent nodes pose a security problem. In the case of the Home Agent, it is reasonable to stipulate that the mobile node and Home Agent should have an IPsec security association, but it is not reasonable to stipulate the same between the Mobile Host and every potential correspondent node. How, then, can a mobile nodes send binding updates to its correspondent nodes securely? The solution adopted in MIPv6 is to use the *return routability procedure.* This allows the mobile node and correspondent node to set up a shared key in a "reasonably" secure manner. The return routability procedure message exchange is shown in Figure 3.

In the return routability procedure, a mobile node sends two messages to the correspondent node, namely the Home Test Init and the Care-of Test Init messages. These messages are sent through the
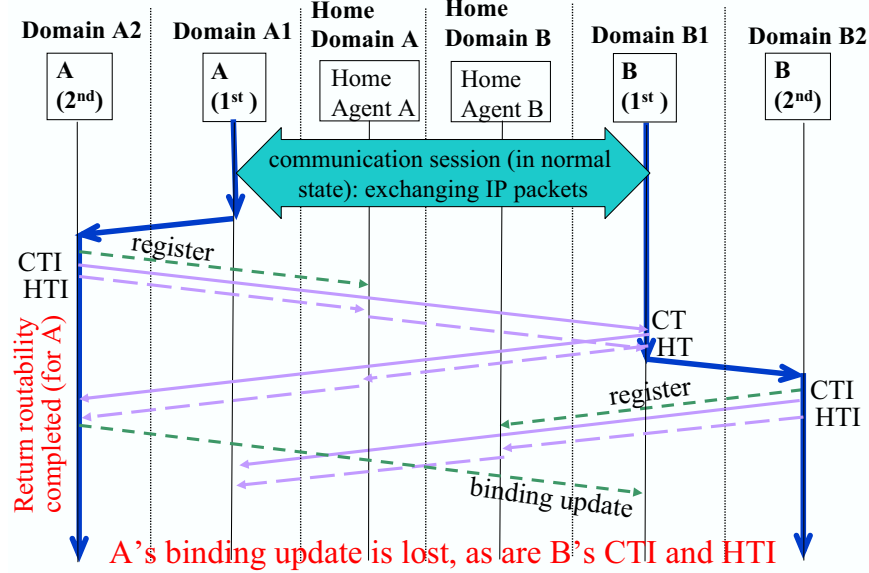
4

Figure 4: One of the Scenarios of Simultaneous Mobility for MIPv6 Illustrated

Home Agent (reversed tunnelled to the Home Agent from the mobile node, and then forwarded to the correspondent node) and directly to the correspondent node, respectively. The correspondent node replies by sending two tokens to the mobile node, one direct to the mobile node addressed to its care-of address (the Care-of Test message), and the other with the home address of the mobile node (the Home Test message). The mobile node needs both tokens to be able to generate the shared key. Thus, the return routability procedure makes reasonably certain that the mobile node is who it claims to be by testing that it is reachable on both the direct path and through its home address. Subsequently, the correspondent node can accept binding updates directly from the mobile node. Because of the use of the return routability procedure, slightly modified versions of Figure 1 and Figure 2 apply to MIPv6. Home agents are found in the two home domains instead of SIP servers. Instead of direct binding updates or re-INVITE messages, three scenarios are possible:

1. Both sides' CTI and HTI messages are lost because of simultaneous mobility. This would look like Figure 2 except that CTI/HTI messages are lost instead of re-INVITES (and home agents are used instead of SIP servers).

2. One side actually completes return routability, but then its binding update is lost because the other side moves. This interesting asymmetric scenario is illustrated in Figure 4.

3. Both sides complete return routability, but then their binding updates are lost due to simultaneous mobility.

2 Analytical Framework

In this section, we introduce a new analytical framework for analysing certain mobility problems such as simultaneous mobility (but not necessarily limited to just analysing simultaneous mobility).

2.1 Fundamental concepts

Definition Two mobile nodes are *in a communications session* if they are actively exchanging data. A communications session may be in a *normal state* or *interrupted state*. It is in a normal state when data
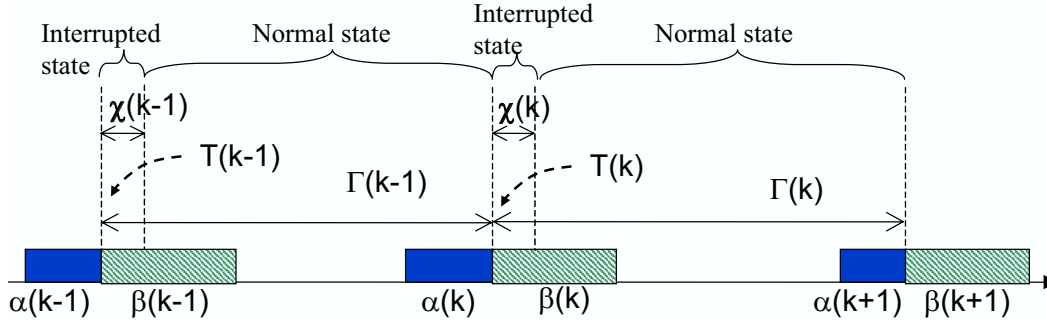
Figure 5: Some of the framework notation is shown in this diagram.

from one node is arriving at the right location for the other node, and vice versa. It is in an interrupted state otherwise.

Example A communications session typically is in an interrupted state from the moment a handoff occurs, until data starts arriving again at the new attachment point (e.g., after a binding update is received at the other node). An illustration of this alternation between normal state and interrupted state is shown in Figure 5.

Remark This is the sort of definition that can get one in trouble – what does "actively" mean? It really depends on what is practically useful. Two nodes could be said to be actively exchanging data if data has been exchanged between them not too long ago. It is very difficult to pin down how long exactly is "not too long ago"; in systems like GPRS, the decision is arbitrarily decided by a timer, where the system switches from an active state to standby state after a timer-based period of inactivity [14]. Another way of looking at it, if there is an application-layer notion like that of SIP session, is to consider the two nodes to be in a communications session, actively exchanging data, as long as the application-layer session is in progress.

2.2 Handoffs and handoff sequences

Definition A *handoff* is a movement of a mobile node from a *previous attachment point* to a *new attachment point*.

Definition The *handoff time* of a handoff is the moment in time when it changes from being reachable at the previous attachment point to not reachable at the previous attachment point. Let the handoff time (of a particular handoff) be T. Then the node needs time for network configuration, so it becomes *reachable* (with an valid IP address at the new network) at time $T + \gamma$. If there is a correspondent node, then some time later, $T + \gamma + \zeta$, it sends a binding update to the correspondent node. The binding update arrives at time $T + \gamma + \zeta + \Delta$. We will use these symbols to represent these differential times after a handoff, in this paper (with subscripts and arguments as necessary – see the definition of handoff sequence shortly). For convenience we may write $\chi(i) = \gamma(i) + \zeta(i) + \Delta(i)$ as we do in Figure 5. So $T(i) + \chi(i)$ is when the binding update arrives at the other node.

Given that time is continuous, we assume that only one handoff can occur at any given moment in time, i.e., handoff times are unique. We note that our definition of handoff and handoff times currently excludes certain types of IP-layer soft handoff (physical layer soft handoff, as in CDMA systems [15], is not a problem for our definition) and bicasting/multicasting schemes. We may expand our framework to include such schemes in the future.

Let A and B be two mobile nodes that are in a communications session with each other, during which they each perform 0, 1 or more handoffs.
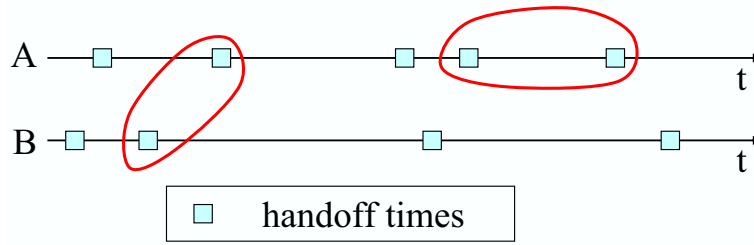
6

Figure 6: Two examples of consecutive handoffs are circled here.

Definition The *handoff sequence* of A is the ordered set

$$\mathbf{H_A} = \{T_A(0), T_A(1), \ldots, T_A(N_A - 1)\} \tag{1}$$

and the *handoff sequence* of B is the ordered set

$$\mathbf{H_B} = \{T_B(0), T_B(1), \ldots, T_B(N_B - 1)\} \tag{2}$$

where $T_A(i)$ is the handoff time of the ith handoff of A (so $T_A(i) < T_A(j) \quad \forall i, j$ such that $0 < i < j < N_A - 1$) and same for B. The function arguments i,j, etc., are the *handoff index number.* In general, when necessary, we will use subscripts to indicate the mobile node and we will show the handoff index number in function arguments.

Definition Two handoffs are *consecutive* (with respect to a pair of mobile nodes) if neither of the mobile nodes performs another handoff in between the two handoffs. For example, if the two handoffs are at A and B, at times $T_A(i_0)$ and $T_B(j_0)$, and suppose that A's handoff is earlier, then saying they are consecutive is equivalent to saying $\{T_A(i) \in \mathbf{H_A} : T_A(i_0) < T_A(i) < T_B(j_0)\} = \oslash$ and $\{T_B(j) \in \mathbf{H_B} : T_A(i_0) < T_B(j) < T_B(j_0) = \oslash$.

As defined, then, consecutive handoffs could be at the same mobile node or at two different mobile nodes. Figure 6 shows two examples of consecutive handoffs, one in which the two handoffs are at different mobile nodes and one in which they are at the same mobile node.

2.3 Binding updates

Definition A binding update is *lost* if it does not arrive at its intended recipient node.

Definition A binding update makes a *belated arrival* if it arrives at a network where the destination address used to be valid for the intended recipient node but is no longer valid (for the intended recipient) at the moment of arrival. For example, if A is the sender and B is the intended recipient, and we are considering the binding update for A's ith handoff, then if B's next handoff is it's jth, then the binding update makes a belated arrival if and only if $T_A(i) + \gamma_A(i) + \zeta_A(i) + \Delta_{A \to B}(i, j) > T_B(j)$

Definition A binding update is *lost through belated arrival* if it makes a belated arrival and is consequently lost.

A node can be lost not necessarily through belated arrival, but through other possible causes of lost binding updates, e.g., network congestion, node failure, link failure, etc. Conversely, a node can make a belated arrival and not be lost, e.g., if there is an agent in that network that can forward the binding update to the current location of the intended recipient (we will introduce such agents in Section 2.4).

Furthermore, we make the following **assumptions** about binding updates:

1. Binding updates cannot and do not contain information about future moves of the sending node.

2. While two nodes are in a communications session, they get information on the location of the other node *only* from binding updates, i.e., they do not actively seek the location of the other node, but only passively accept binding updates.

3. Unless otherwise stated, a binding update is sent directly to the most current known address (i.e., known by the sender) of the intended recipient.

4. Regarding the relative timings of binding update latencies and consecutive handoffs of a receiving mobile node, the timescale of the latencies for binding updates to arrive is assumed much smaller than the average inter-handoff time. In other words, $\Delta \ll E\{T(i+1) - T(i)\}$, where $E\{\bullet\}$ denotes expectation.

5. Following up from assumption 4, we assume that it is extremely unlikely[3] that a binding update would be sent and the recipient moves *twice* before the binding update arrives at the previous network of the recipient.

6. Following up on the previous assumption, we also assume that if there is a forwarding location proxy in the previous network of the recipient, that it will correctly forward the binding update to the recipient, which would only have *moved once* from the previous network.

2.4   Location proxies and binding update proxies

Here we introduce two kinds of proxies. These proxies, if used carefully, can help prevent the simultaneous mobility problem. These proxies are abstract proxies – the definitions are more about network functionality than specific implementations as network elements. Thus, we will see how familiar network elements like Home Agents can be described as having certain proxy functions, or can be enhanced for such purposes. The abstraction of these proxies will allow general problems and solutions (related to simultaneous mobility) to be discussed without being unnecessarily bogged down by details of specific mobility protocols. We expect that these proxies should be stationary, not mobile, to avoid introducing unnecessary complications.

Definition A *location proxy* (of a mobile node) is a network function that is used to locate the mobile node. A *forwarding location proxy* will forward messages (including binding updates) to the most recent location that it knows for the mobile node. A *redirecting location proxy* will redirect (e.g., by responding to a query with the latest address) messages to the most recent location that it knows for the mobile node. An *intercepting location proxy* intercepts, and may act on (forwards or redirects), messages in packets not addressed to it. A *non-intercepting location proxy* only acts on messages in packets addressed to it.

The differences between the types of proxies are shown in Figure 7. It is important to note that whereas a forwarding location proxy will pass along messages toward the final destination, a redirecting location proxy will not do this, but just return location information that can be used to send the message toward the final destination. Note also that our definitions here are not specific to any particular mobility protocol. We will give examples related to specific mobility protocols shortly, after introducing a few more terms.

Definition A location proxy is *up-to-date* with respect to a particular mobile node, if that mobile node continually updates the proxy with its latest address after each move.

---

[3]Extending our analysis to include this kind of very rare occurrence would complicate matters, and obscure our understanding of the great majority of cases. We follow the example of typical analyses of the Decision Feedback Equalizer (in Digital Communications) where it is assumed that in the great majority of cases the decision feedback is correct. Exact analysis of error propagation is possible but less useful.
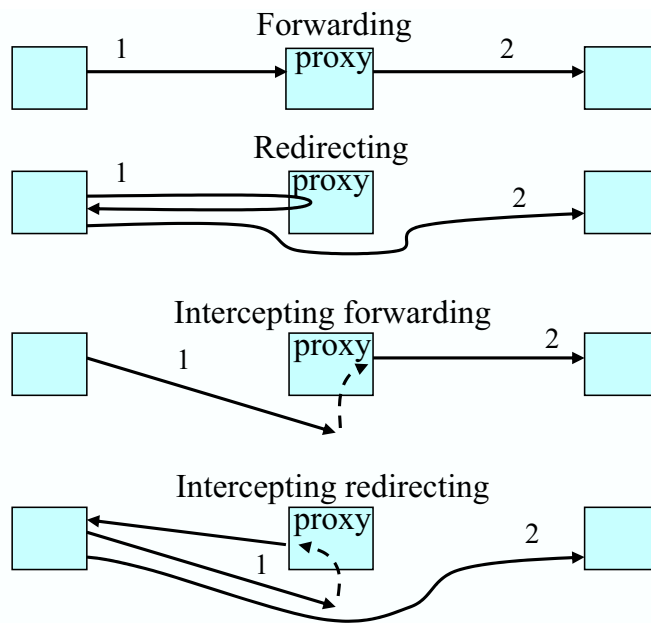
Figure 7: Four types of proxies are shown here.

Definition A *pro-active* location proxy keeps a copy of mobility related signalling messages (typically, binding updates, but possibly other messages like Care-of-Test Init, when procedures like the return routability procedure are used before the binding update is sent). It keeps the messages for a short while, τ, after receiving and acting on them (i.e., after redirecting and/or forwarding the message). The messages are kept in the *location proxy cache* and indexed by destination node. The messages are discarded after time τ has elapsed. If during this period of time, the pro-active location proxy receives a binding update from any one of the destination nodes in its location proxy cache, it either (a) redirects to the new address (if it is a redirecting pro-active location proxy); or (b) forwards the corresponding saved message(s) to it (if it is a forwarding pro-active location proxy).

Example The MIP Home Agent is a forwarding location proxy (of the intercepting kind). DNS servers are non-intercepting redirecting location proxies. MIP-LR Home Location Registers are non-intercepting redirecting location proxies. SIP servers are non-intercepting proxies that can be either forwarding location proxies (known as *proxy servers* in SIP terminology [16]) or redirecting location proxies (known as *redirect servers* in SIP terminology [16]). Except for DNS servers, the other examples given here are typically used in mobility schemes as up-to-date location proxies. In TCP Migration [10], though, DNS servers are part of the mobility scheme, and so they are up-to-date location proxies in that scheme. Most existing location proxies are not pro-active location proxies. However, pro-active location proxies may be useful in solutions to the simultaneous mobility problem as will be discussed in Sections 4 and 5.

In this paper we don't emphasize the difference between forwarding/redirecting of *signalling* and forwarding/redirecting of *data* because we are more interested in the signalling in this paper. We assume that if the mobility signalling gets to its intended recipient, the mobility schemes should, and must, take care of the data traffic correctly. In some cases, e.g., with Mobile IP, the Home Agent forwards both signalling and data, whereas in other cases, e.g., MIP-LR and SIPMM, the location registers or SIP servers are only involved in the signalling.

Definition A *binding update proxy* acts on behalf of a mobile node to send its binding updates to its correspondent nodes' latest addresses. It would typically engage the services of a location proxy of each correspondent node either for redirection to the correspondent node's latest address, or for forwarding the

relevant binding update accordingly. At the same time, it would also forward a copy of the message to the latest address it knows for the correspondent node. A mobile node on whose behalf a binding update proxy acts may be referred to as a *master* of that binding update proxy.

Why can't a mobile node serve as its own binding update proxy? The difference is that the mobile node is mobile whereas the binding update proxy is stationary. It is a fixed node that can obtain the latest address of any correspondent nodes. One problem with binding update proxies, as defined above, is that the correspondent node may move and its location proxy may only receive this information *just after* responding to a query or forwarding a binding update from the binding update proxy. A pro-active binding update proxy, which we will now define, might work with a redirecting pro-active location proxy to handle the situation.

Definition A *pro-active binding update proxy* not only queries for the latest addresses of the correspondent nodes of its master(s); it also keeps the binding updates for a short while, ρ, after receiving and forwarding them. The messages are kept in the *binding update proxy cache* and indexed by destination node. The messages are discarded after time ρ has elapsed. If during this period of time, the pro-active binding proxy receives a redirection regarding any one of the destination nodes in its binding update proxy cache, it forwards the corresponding saved binding update(s) to it.

3 Analysing the Simultaneous Mobility Problem

3.1 On the loss of binding updates and occurrence of the simultaneous mobility problem

Definition The *simultaneous mobility problem* occurs when there are two mobile nodes in a communications session in normal state, and they both move such that the binding updates that they send to each other are both lost through belated arrival, and such that the communications session never returns from interrupted state to normal state, but is ended.

We now prove four lemmas that cover the two cases of what happens when there is a pair of handoffs at two mobile nodes, and the binding update from the earlier one arrives (a) later than the time the other node moves (Lemmas 3.1 and 3.2); or (b) earlier than the time the other node moves (Lemmas 3.4 and 3.5).

Lemma 3.1 *Given a pair of consecutive handoffs, one for each of two mobile nodes in a communications session in normal state (up till the first handoff). In the absence of location proxies for either mobile node (or there might be location proxies but they are not used/involved), any binding update sent by the earlier moving mobile node will be lost through belated arrival, if and only if the binding update doesn't arrive at the other mobile node before it moves.*

Proof Suppose without loss of generality that A moves before B. Let the handoff times be $T_A(i_0)$ and $T_B(j_0)$, so $T_A(i_0) < T_B(j_0)$. Since A and B are in a communications session in normal state up till $T_A(i_0)$, then up till $T_A(i_0)$, anything sent by A arrives at B and vice versa. Since the two handoffs are consecutive, then by definition there is no other handoff in the time interval $[T_A(i_0), T_B(j_0)]$. By our third assumption on binding updates, A's binding update would be addressed to the latest address it has for B. So for time interval $[T_A(i_0), T_B(j_0)]$, anything sent by A will still be addressed to B's pre-handoff address, and still arrives at B, including A's binding update. However, as soon as $t \geq T_B(j_0)$, B would no longer be reachable at its pre-handoff address. In some scenarios, a location proxy for B would be able to prevent the binding update from being lost. However, in the absence of a location proxy, the binding update would just go to B's previous address and disappear there. Hence, it would be lost through belated arrival.

Conversely, suppose A's binding update is lost through belated arrival. As shown, for time interval $[T_A(i_0), T_B(j_0)]$, anything sent by A will still be addressed to B's pre-handoff address, and still arrives at
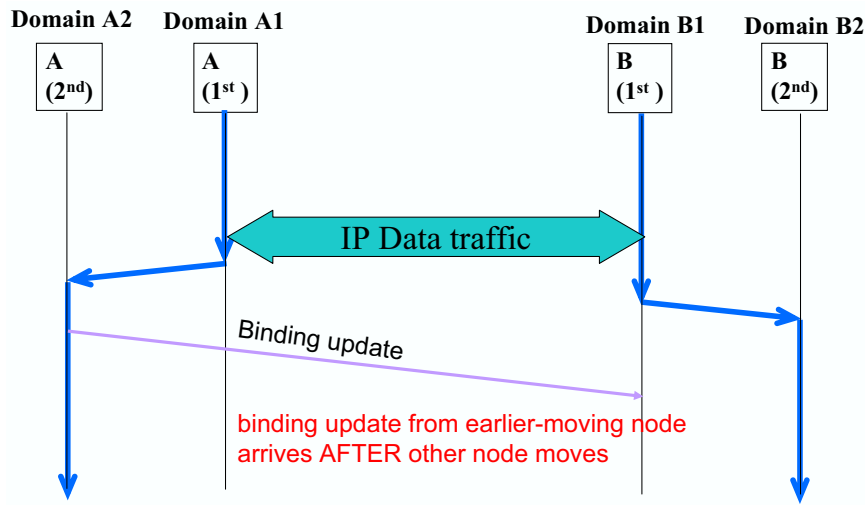
Figure 8: The situation considered by Lemmas 3.1 and 3.2.

B, including A's binding update. So if it arrives before $T_B(j_0)$, it will not be lost through belated arrival. Thus, A's binding update cannot arrive before $T_B(j_0)$. Therefore it arrives after B has moved. ∎

This lemma and the next are making assertions about cases where the binding update from the earlier-moving mobile node arrives after the later-moving mobile node has moved. This is shown in Figure 8.

Lemma 3.2 *Given a pair of consecutive handoffs, one for each of two mobile nodes in a communications session in normal state (up till the first handoff). In the absence of location proxies for either mobile node (or there might be location proxies but they are not used/involved), the simultaneous mobility problem will occur, if and only if the binding update sent by the earlier moving mobile node doesn't arrive at the other mobile node before it moves.*

Proof We use A and B as the first and second moving nodes again. If the binding update from A doesn't arrive at the other node before it moves, then by Lemma 3.1, it is lost through belated arrival. Then, at time $T_B(j)$, B doesn't have A's new address. Since A's binding update is lost, by the time B is sending it's binding update (i.e., $T_B(j) + \lambda_B(j) + \zeta_B(j)$) it will send it to A's previous address. Thus, in the absence of location proxies, B's binding update will also be lost through belated arrival. So both A's and B's binding updates are lost through belated arrival. By definition, the simultaneous mobility problem has occurred.

Conversely, if the simultaneous mobility problem has occurred, then by both binding updates are lost through belated arrival, so A's binding update is lost through belated arrival. ∎

From the above proof, the following corollary emerges.

Corollary 3.3 *Given a pair of consecutive handoffs, one for each of two mobile nodes in a communications session in normal state (up till the first handoff), In the absence of location proxies for either mobile node (or there might be location proxies but they are not used/involved), if the binding update from the node that moved first is lost through belated arrival, the binding update from the node that moved second will also be lost through belated arrival.*

Lemma 3.4 *Given a pair of consecutive handoffs, one for each of two mobile nodes in a communications session in normal state (up till the first handoff), the simultaneous mobility problem does not occur if the binding update from the node that moved earlier reaches the other node before that node moves.*
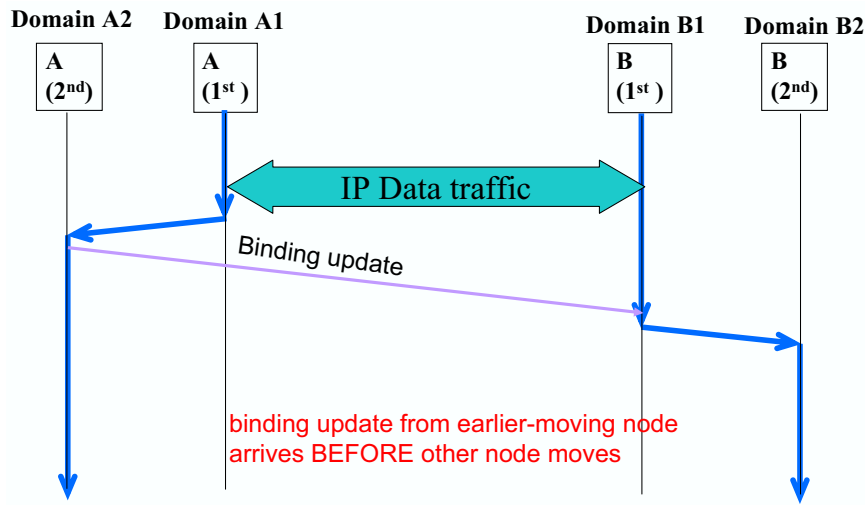
Figure 9: The situation considered by Lemmas 3.4 and 3.5.

Proof As in the proof of Lemma 3.1, we can argue that for time interval $[T_A(i_0), T_B(j_0)]$, anything sent by A still arrives at B, including A's binding update. Thus, B can then send its binding update correctly to A's new address after it moves at $T_B(j_0)$. Therefore, the simultaneous mobility problem does not occur. ▪

This lemma and the next are making assertions about cases where the binding update from the earlier-moving mobile node arrives before the later-moving mobile node has moved. This is shown in Figure 9. NB: the converse of Lemma 3.1 is not necessarily true, i.e., we cannot say that if the simultaneous mobility problem does not occur, then the binding update from the node that moved earlier reaches the other node before that node moves. The reason this is not necessarily true is that location proxies could be used, as we will demonstrate later. However, we first extend Lemma 3.4 to the case that location proxies are excluded, where we can make a stronger statement.

Lemma 3.5 *Given a pair of consecutive handoffs, one for each of two mobile nodes in a communications session in normal state (up till the first handoff). In the absence of location proxies for either mobile node (or there might be location proxies but they are not used/involved), the simultaneous mobility problem does not occur* if and only if *the binding update from the node that moved earlier reaches the other node before that node moves.*

Proof This has been partially proved in the proof for Lemma 3.4. What remains is to prove that if the simultaneous mobility problem does not occur, the binding update from the node that moved earlier reaches the other node before that node moves. Supposing the simultaneous mobility problem does not occur. That means both binding updates arrive at the other node. B's binding update therefore cannot be lost through belated arrival, so B must have successfully received A's binding update. By the 3rd assumption on binding updates, A's binding update could not have been addressed to B's new location since A moved first. Given that location proxies are not used, there is no way that B could successfully receive A's binding update after $T_B(j_0)$. Therefore, A's binding update must have reached B before $T_B(j_0)$, i.e., before B moved. ▪

Remark What about if A's binding update successfully reaches B before it moves, but B's binding update does not reach A because A's next handoff happens before the arrival of B's binding update? Does Lemma 3.4 break down? No – in that case the lemma, applied to $H_A(i_0)$ and $H_B(j_0)$ would correctly show that the problem is not between those two handoffs, but applied to $H_B(j_0)$ and $H_A(i_0 + 1)$ would correctly show that the simultaneously mobility problem occurs then, with B as the earlier moving node.

3.2 Analysis of specific mobility protocols

Theorem 3.6 *The simultaneous mobility problem does not occur in MIP (without Route Optimization)*

Proof No binding updates are sent directly to mobile nodes. So, there are none to be lost through belated arrivals. Instead, Home Agents serve as up-to-date intercepting location proxies that forward data to the right location as soon as Mobile IP registration occurs after each handoff. ■

In MIPv4 with route optimisation (MIP-RO), after a node has moved from a previous network attachment point, it uses the Previous Foreign Agent Notification Extension (PFANE) to inform the previous Foreign Agent about its new location. So the previous Foreign Agent serves as an up-to-date forwarding location proxy (only for that particular handoff – once the mobile node moves again, the current Foreign Agent becomes the previous Foreign Agent and assumes this role). The second response to receiving packets addressed to old care-of address of the mobile node is that the previous Foreign Agent will also send a Binding Warning message to the Home Agent so the Home Agent then issues a Binding Update to the Correspondent Node. We can think of this second response as the Foreign Agent not only Serving as an up-to-date forwarding location proxy, but also as an up-to-date redirecting location proxy (this second function in conjunction with the Home Agent).

We will prove Theorem 3.8 that the simultaneous mobility problem doesn't occur in MIP-RO. However, it is instructive to first consider the following Lemma 3.7.

Lemma 3.7 *If we remove the Binding Warning feature, the simultaneous mobility problem will occur in MIP-RO if and only if the binding update from the earlier moving node arrives at the later moving node's previous Foreign Agent after that node has moved but before that Foreign Agent receives the PFANE.*

Proof Consider a pair of consecutive handoffs, one for each of two mobile nodes in a communications session in normal state (up till the first handoff). Suppose without loss of generality that A moves before B. Let the handoff times be $T_A(i_0)$ and $T_B(j_0)$, so $T_A(i_0) < T_B(j_0)$. Then, A's binding update either arrives at B before or after $T_B(j_0)$. If it arrives before $T_B(j_0)$, then by Lemma 3.4 the simultaneous mobility problem does not occur. If it arrives after $T_B(j_0)$, it would be a belated arrival, as B would have moved. Here, there are two sub-cases:

- A's binding update arrives after B's previous Foreign Agent receives the PFANE.
- A's binding update arrives before B's previous Foreign Agent receives the PFANE.

In the first sub-case, by assumptions 4 to 6 on binding updates, A's binding update is forwarded correctly from B's previous network to B, by the previous Foreign Agent. Meanwhile, B's binding update may arrive at A's previous Foreign Agent, so by the same reasoning it is forwarded to A's new location. Thus, the simultaneous mobility problem does not occur. In the second sub-case, A's binding update is lost through belated arrival. Could B's binding update also be lost? Yes. Even though it may be highly unlikely, B's binding update *may* arrive before A's PFANE at A's previous Foreign Agent. By definition, B's binding update would be lost through belated arrival. Thus, by definition, the simultaneous mobility problem occurs. This scenario is shown in Figure 10.

Conversely, supposing the simultaneous mobility problem occurs, then both binding updates must have arrived before the other node's previous Foreign Agent receives the PFANE but after the other node moved. Otherwise, if it arrived before the other node moved, then by definition simultaneous mobility does not occur. Or supposing A's binding update arrives after B's previous Foreign Agent receives the PFANE. Then B will receive A's binding update and can send packets correctly to A (whereas A's packets get forwarded by B's previous Foreign Agent and still arrive correctly at B).

Theorem 3.8 *The simultaneous mobility problem does not occur in MIPv4 with Route Optimization.*
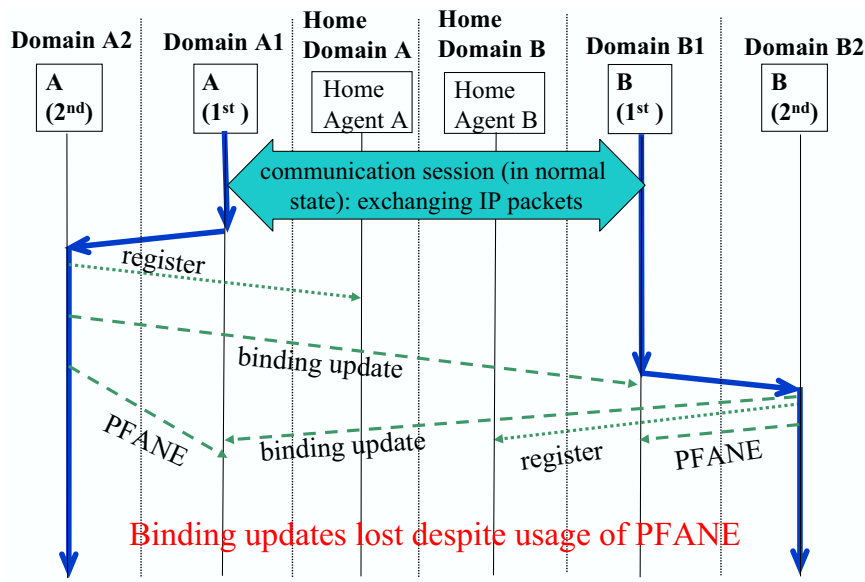
13

Figure 10: This flow diagram shows how both binding updates can be lost even if forwarding from the previous Foreign Agent is in place.

Proof As we have seen in the proof of Lemma 3.7, when Binding Warnings are not used, the simultaneous mobility problem occurs only in the event that both binding updates arrive after the destination node has moved but before the destination node's previous Foreign Agent has received the PFANE. In the case that Binding Warnings are used, then when each of the PFANEs arrives at their respective previous Foreign Agents any subsequent packet from the other node will trigger a Binding Warning, and the Home Agent will take care of sending a Binding Update to the other side. Thus, each node quickly gets an up-to-date Binding Update from the other side. Therefore, there is no simultaneous mobility problem.

Theorem 3.9 *The simultaneous mobility problem may occur with the regular version of SIP mobility, MIP-LR and MIPv6 with Route Optimization.*

Proof In the regular version of these protocols, once a communications session is in progress, binding updates are sent directly to correspondent nodes without the assistance of location proxies. Supposing there are a pair of consecutive handoffs of A and B, at times $T_A(i_0)$ and $T_B(j_0)$, where the communications session is in normal state up to $T_A(i_0)$ and $T_A(i_0) < T_B(j_0)$. By definition, the binding update from A arrives at B at time $T_A(i_0)+\gamma(i_0)+\zeta(i_0)+\Delta_{A\rightarrow B}(i_0)$. For MIP-LR and SIP, $\Delta_{A\rightarrow B}(i_0)$ is the latency for the binding update to reach B. For MIPv6 with Route Optimization, it also includes to time for performing the return routability procedure. If

$$T_A(i_0) + \gamma(i_0) + \zeta(i_0) + \Delta_{A\rightarrow B}(i_0) > T_B(j_0) \tag{3}$$

then A's binding update is lost through belated arrival. There is nothing to prevent Equation 3 from being true for any given pair of consecutive handoffs of A and B. Thus, by Lemma 3.2, the simultaneous mobility problem may occur. ■

3.3 Probability of occurrence of simultaneous mobility

In [12] we introduced a simple mathematical model for estimating the probability of occurrence of simultaneous mobility. In this paper, we will not develop this model beyond what we had presented before, but here we summarize it for completeness.

Consider two consecutive handoffs, one each at mobile nodes A and B. According to Lemma 3.2, the simultaneous mobility problem occurs if and only if the binding update from the earlier moving node

arrives after the other node has moved. Mathematically, this is written as $T_A + \gamma + \zeta + \Delta_{A \to B} > T_B$ (if A is the earlier moving node) or $T_B + \gamma + \zeta + \Delta_{B \to A} > T_A$ (if B is the earlier moving node). Putting the two inequalities together, then, we have

$$T_A - \alpha < T_B < T_A + \beta \tag{4}$$

where $\alpha = \gamma + \zeta + \Delta_{B \to A}$ and $\beta = \gamma + \zeta + \Delta_{A \to B}$ are convenient short forms. Then we define the concept of "vulnerability interval" as $\beta + \alpha$, which is the time around a handoff during which the two mobile nodes are vulnerable to the simultaneous mobility problem if another handoff occurs at the other mobile node.

It is reasonable to model the handoff times for A and B as independent Poisson processes. In this model, the intervals between consecutive handoffs at A, $\Gamma_A(k-1) = T_A(k) - T_A(k-1)$, $\Gamma_A(k) = T_A(k+1) - T_A(k)$, etc., are independent exponentially distributed random values, and similarly for the corresponding intervals between consecutive handoffs at B. Then it is easy to argue that the probability of the simultaneous mobility problem occurring can be estimated as

$$P_0 \approx \frac{E(\alpha + \beta)}{E(\Gamma)} \tag{5}$$

If there are N handoffs occurring at each of the two mobile nodes, it was proposed [12] that the probability of the simultaneous mobility problem occurring could be estimated by $P_N = 1 - (1 - P_0)^N$.

4   Solution Mechanisms

A few previous papers have hinted at the benefits of some kind of proxies in fixed locations to enable "communication to continue even when both end-hosts move simultaneously" [17]. However, as far as we know, previous work has not analysed the problem to the level we have in Section 3 nor has a systematic analysis of solutions applied to a range of mobility protocols been previously provided, with the exception of [12]. Nevertheless, our understanding of the solutions has improved since we wrote [12], and we present here (in this section and the next section) our latest analysis.

Solution Mechanisms			MIP-RO	MIPv6	SIPMM	MIP-LR
Receiver-side Mechanisms	At previous network (location proxies)	Retransmission			possible[a]	
		Forwarding	yes			
		Pro-active forwarding				
		Redirecting	yes			
		Pro-active redirecting				
	At home network of receiver (location proxies)	Retransmission			possible[a]	
		Forwarding	yes[b]	yes[b]	possible[c]	
		Pro-active forwarding				
		Redirecting				yes[b]
		Pro-active redirecting				
Sender-side Mechanisms	At home network of sender (binding update proxies)	Retransmission			possible[a]	
		Forwarding			possible[c]	
		Pro-active forwarding				
		Redirecting				
		Pro-active redirecting				
	At sender	Retransmission			yes	

[a]If stateful SIP server used with Record Route
[b]Often or always unused in mid-session
[c]If Record Route is used

We start by examining solution mechanisms in this section. Solution mechanisms are specific mechanisms/functions that could be used (typically in conjunction with other mechanisms/functions) in a solution for the simultaneous mobility problem for a given mobility protocol. We divide them broadly into receiver-side and sender-side mechanisms, where receiver and sender are with respect to the sending of a particular binding update. Receiver-side mechanisms typically can be found in the previous network or home network of the receiver and act on behalf of a receiver to help it be reached. We explore these mechanisms in Section 4.2. Sender-side mechanisms typically can be found in the home network of the sender, or in the sender itself, and act on behalf of the sender to try to reach the receiver. We explore these mechanisms in Section 4.3. The receiver may be moving simultaneously with the sender and may not receive the binding update if none of these mechanisms are used. But first we briefly consider an alternative approach using soft handoffs.

4.1 Soft handoff approach

Suppose a mobile node can have more than one valid IP address. This is sometimes referred to as "simultaneous mobility bindings", and should not be confused with the simultaneous mobility problem. We call it the *soft handoff approach*, since it is similar to soft handoffs in CDMA mobile systems [15]. The idea is that if the previous IP address and new IP address can both be used to reach the mobile node during the time around a handoff, that may help solve the simultaneous mobility problem. Binding updates sent to the previous IP address would arrive correctly. However, this is not a universally applicable solution, for the following reasons:

- The radio network technology needs to be able to support multiple concurrent IP addresses for the wireless interface(s). It is too much to require this for solving simultaneous mobility.

- Resource utilization is not efficient, because of redundant allocation of resources (on both branches) during the period of simultaneous mobility bindings.

- It is unclear how long the simultaneous mobility bindings need to be active to ensure that no problems will occur with simultaneous mobility – how long should it wait for a possibly delayed binding update? The longer it waits, the more the scheme uses valuable network resources redundantly.

Since our solution must work for any radio network technology, use of simultaneous bindings is not a satisfactory solution. Hence we will not examine mechanisms related to the soft handoff approach in this paper.

4.2 Receiver-side mechanisms

4.2.1 Timer based retransmissions

One could imagine a forwarding location proxy automatically retransmitting a binding update if it has not gotten confirmation that the binding update was successfully received by the intended receiver. This location proxy could be located in the receiver's home network or a visited network (e.g., the previous network or latest network).

Location proxies that retransmit based on timeouts are similar to pro-active location proxies in that both need to store the message briefly, to retransmit if necessary. The difference is in the conditions for retransmission. A pro-active proxy retransmits as soon as a new address is obtained, whereas a timer-based retransmission may be too slow.

Existing implementation   A stateful SIP server [16] could retransmit binding updates (re-INVITES) after the expiration of a timer. This could be located in the home network of the receiver or the visited network of the receiver. In order to ensure that the re-INVITE message (and other signalling) goes through this server, the Record-Route option could be used in the initial INVITE message to add the server to the signalling path.

4.2.2 Forwarding (regular, passive type)

Forwarding mechanisms on the receiver side allow binding updates to be forwarded from a location proxy in the previous network to the correct new location of the receiver. Forwarding mechanisms from a previous network may also forward data packets (since it is forwarding packets, anyway, it might as well forward data packets as well). One could also imagine such a location proxy in the receiver's home network as well.

Existing implementation in the receiver's previous network In MIP-RO, the previous Foreign Agent serves this role (see Section 3). It also forwards data packets. Unfortunately, this ability is missing from MIPv6, perhaps because no Foreign Agents are used in MIPv6 (and so, there is no natural forwarding agent present in the previous network). Thus, the problem of simultaneous mobility remains in MIPv6. Similarly, SIPMM and MIP-LR lack such functionality.

Existing implementation in the receiver's home network An example is the Home Agent in MIP and MIPv6. A SIP server in the receiver's home network could also serve in this capacity, e.g., if it places itself in the signalling path using the Record Route field in the initial INVITE message.

4.2.3 Pro-active forwarding

Regular, passive forwarding may be insufficient to solve the simultaneous mobility problem. A pro-active forwarding location proxy may help in some solutions.

4.2.4 Redirecting

Redirecting mechanisms on the receiver side can help to get messages like binding updates to the right place.

Existing implementation in the receiver's previous network In MIP-RO, the previous Foreign Agent serves this role (see Section 3).

Existing implementation in the receiver's home network In MIP-LR, the HLR does this, but only before a communications session begins. Then, it is not involved in control signalling during the communications session. Thus, it does not count as a proper implementation of a solution mechanism for the simultaneous mobility problem

4.2.5 Pro-active redirecting

Regular redirecting may be insufficient to solve the simultaneous mobility problem. A pro-active redirecting location proxy may help in some solutions.

4.3 Sender-side mechanisms

4.3.1 Timer based retransmissions

One could imagine a forwarding location proxy automatically retransmitting a binding update if it has not gotten confirmation that the binding update was successfully received by the intended receiver. This location proxy could be located in the sender's home network or even the sender itself (for end-to-end retransmission).

Existing implementation A stateful SIP server could retransmit binding updates (re-INVITES) after the expiration of a timer. This could be located in the home network of the receiver or the visited network of the receiver. In order to ensure that the re-INVITE message (and other signalling) goes through this server, the Record-Route option could be used in the initial INVITE message to add the server to the signalling path.

4.3.2 Forwarding (regular, passive type)

Forwarding mechanisms in the sender's home network can help to get messages like binding updates to the right place, but are probably less useful than those on the receiver side.

4.3.3 Pro-active forwarding

Regular, passive forwarding may be insufficient to solve the simultaneous mobility problem. A pro-active binding update proxy may help in some solutions, where it attempts to find the most current location of the receiving node and re-try the forwarding there.

4.3.4 Redirecting

Redirecting mechanisms in the sender's home network can help to get messages like binding updates to the right place, but are probably less useful than those on the receiver side.

4.3.5 Pro-active redirecting

Regular redirecting may be insufficient to solve the simultaneous mobility problem. A pro-active redirecting location proxy may help in some solutions.

5 Solutions

Here we suggest combinations of solution mechanisms to form complete solutions to the simultaneous mobility problem, for specific mobility protocols, namely MIPv6, MIP-LR and SIPMM. Our choice of preferred solution for each protocol is guided by a desire to minimize changes to the characteristics and flavor of the protocol.

5.1 MIPv6

Three solutions are considered here:

Forwarding proxy in previous network. For MIP-RO, as described earlier, Foreign Agents in the previous network act as forwarding proxies. For MIPv6, however, Foreign Agents are not used. Thus, ordinary routers in the previous network would need to be augmented with forwarding proxy functionality. This would involve significant challenges and modifications to MIPv6. For example, a mechanism would be needed to securely update the router with the latest IP address of the mobile node. However, getting ordinary IPv6 routers to do this kind of forwarding might be considered an unacceptable demand, and some kind of agent might need to be introduced.

Combination of sender-side and receiver-side mechanisms. A combination of sender-side pro-active binding update proxies and receiver-side pro-active redirecting location proxies was proposed as a general solution in [12]. In [12], the solution was applied to SIPMM and MIP-LR. Here we propose that it can also be applied to MIPv6. The Home Agents of the sender and receiver, respectively, can server as
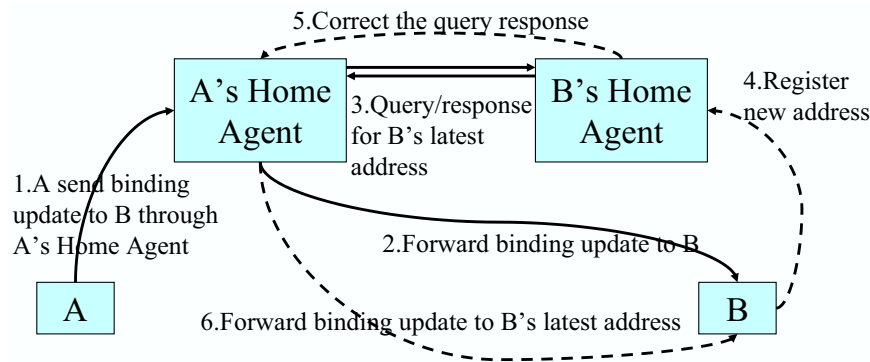
Figure 11: A combination of sender-side and receiver-side mechanisms to prevent the simultaneous mobility problem from occurring in MIPv6.

the pro-active binding update proxy and pro-active redirecting location proxy. Return routability has to be modified so the CTI message goes through the sender's Home Agent. Another modification to MIPv6 is that the binding update must first be reversed tunnelled to a mobile node's own Home Agent before being forwarded to the correspondent node.

The revised MIPv6 update procedure would work as follows. Let the two mobile nodes be A and B as usual. A sends its CTI and binding update messages to B through A's Home Agent, rather than directly to B's care-of address. However, A's Home Agent will then forward these messages to B at B's care-of address. A's Home Agent, acting as a pro-active binding update proxy will also keep a copy of any such message for a period τ. It would then query B's Home Agent (a pro-active location proxy for B) to find out if B has a newer address. B's Home Agent responds immediately but keeps a copy of the query for a period ρ. If before this period is over, B's Home Agent receives a registration for B at a new address, it pro-actively corrects its query. A's Home Agent will then forward the message to the new address. This solution is shown in Figure 11.

The selection of τ and ρ should be carefully chosen based on reasonable estimates of the appropriate signalling and computational delays of the network. It is clear that $\tau > \rho$ so A's Home Agent can respond to any query response correction from B's Home Agent.

Receiver-side mechanisms only. The two solutions discussed so far are not good in that they require significant changes to MIPv6. A more MIPv6-centric solution is preferable. We therefore consider a solution where just the receiver's Home Agent is involved. We let A be the sender (of the CTI, HTI or binding update). A sends all these control messages to B using B's *home address*, thus forcing B's Home Agent to be involved. B's Home Agent will act as a pro-active forwarding location proxy (a slight modification from its usual role as a forwarding location proxy), forwarding the control message to B as usual, but keeping a copy of it for time τ. If it gets any binding updates from B during that time, it pro-actively forwards the message to B. This solution is shown in Figure 12.

We note that the main modification to Home Agents implementing this solution is becoming pro-active forwarding location proxies, not just forwarding location proxies. The main modification to mobile nodes implementing this solution is also small – to send the CTI and binding update to the home address of the correspondent node instead of directly to its care-of address.

Evaluation. It is clear that the 3rd solution, with receiver-side mechanisms only (sending messages to the other node's Home Agent) is the cleanest solution with the least changes to MIPv6. The adding (in the 2nd solution) of a query/response capability to Home Agents is quite a drastic change that is out of character for MIPv6. After the removal of Foreign Agents in going from MIP to MIPv6, the adding of a forwarding proxy in the previous network with substantial functionality (in the 1st solution) is not
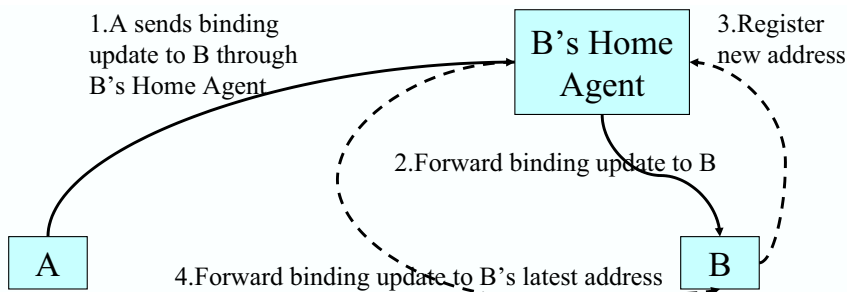
Figure 12: Using just receiver-side mechanisms to prevent the simultaneous mobility problem from occurring in MIPv6.

desirable.

5.2 MIP-LR

Forwarding proxy in previous network.   There are two varieties of MIP-LR in the literature: one with Foreign Agents [3], and one without [2]. The version without Foreign Agents uses Advertisement Agents.

For the purposes of placement of the Interceptor function, it doesn't matter whether Foreign Agents or Advertisement Agents are in use. The point is that there is some kind of agent in each of the foreign networks, and the Interceptor function can be placed here. MIP-LR would need modification so the mobile node sends a binding update to the Foreign Agent (or Advertisement Agent) in the previous subnet as soon as it obtains its new IP address.

Sender-side and receiver-side mechanisms.   For this solution, the binding update sent by a mobile node to its HLR has a list of correspondent nodes and their addresses. The HLR, which already anyway performs the role of a redirecting location proxy, is enhanced to be a pro-active redirecting location proxy. It also acts as a pro-active binding update proxy, since it anyway obtains the current binding information as part of MIP-LR updating after each handoff. In order to do this, the HLRs must be enhanced to pro-actively retransmit binding updates and to query other HLRs for correspondent nodes' addresses. In order to minimize changes to MIP-LR, the HLR-initiated binding updates are only sent when necessary, i.e., when the queries return a newer address for a correspondent node than the one provided by the mobile node. The solution is illustrated in Figure 13.

Evaluation.   We don't consider a solution using only receiver-side mechanisms as we did for MIPv6. This is because the HLR would then have to become a pro-active forwarding proxy. Such a change is too much for MIP-LR, one of whose points is that no forwarding location proxies are used (but multiple replicated HLRs are used).

Of the two solutions considered, the one where the HLRs are enhanced is recommended, and is the one we introduced in [12].

5.3 SIPMM

SIP allows much flexibility in placement and usage of SIP servers in the signalling path between two mobile nodes. The Record-Route field is an optional field in the SIP header that allows SIP servers to remain in the signalling path between two SIP end-nodes during a communications session. To apply our principle of selecting solutions that minimize changes to the character/flavor of the mobility protocols, we need to first select a common "typical" usage scenario – we assume that often there will be a SIP server in each mobile node's home network that serves as an up-to-date location proxy for it.
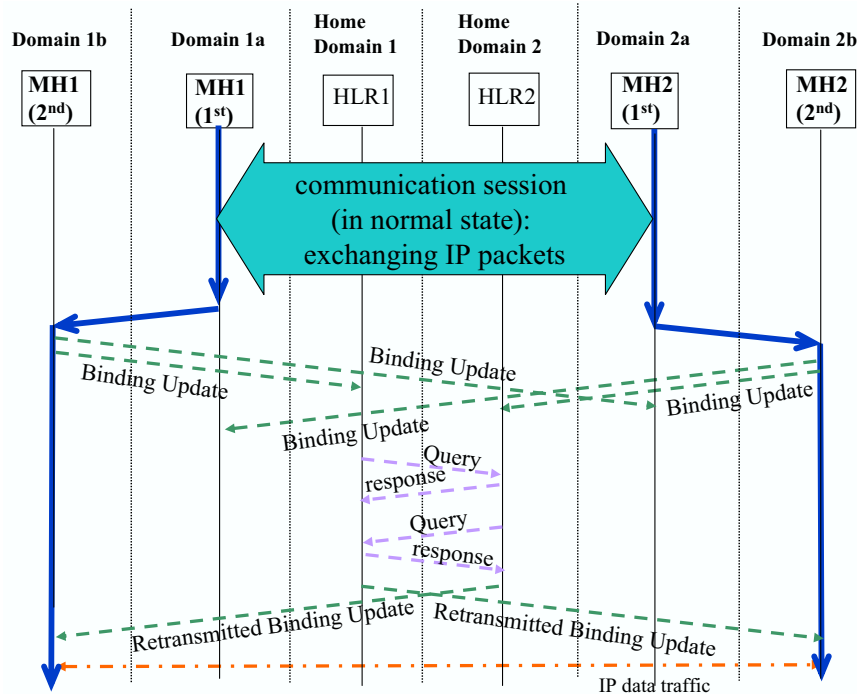
Figure 13: A solution to prevent the simultaneous mobility problem from occurring in MIP-LR.

Forwarding proxy in previous network. Like the first proposed solution for MIPv6, we consider adding a forwarding location proxy to the previous network of a mobile node.

For SIPMM, the most natural location of the forwarding proxy might be in an RTP (Real-time Transport Protocol) translator [18], since these are already used to forward traffic, among other things. Why not put a SIP server in the previous network for this role? One problem is that if we keep adding SIP servers in previous networks to the signalling path (using Record-Route), the signalling path becomes inefficient as the mobile node moves and the signalling path goes through more and more SIP servers in previous networks. Note that the existing RTP translator only forwards data traffic from the previous network, so we are proposing an extension to RTP translators both to intercept and to forward SIP signalling as well. For receiving update signalling, the RTP translator can be enhanced to act as a SIP registrar.

Receiver-side mechanisms. For SIPMM, the receiver-side home network SIP server already has some location proxy functionality, and can be converted to a pro-active one. We first consider the case that it acts as a forwarding location proxy (it can also be a redirecting location proxy, which we will consider in the next paragraph). The SIP server immediately retransmits the re-INVITE upon receiving a REGISTER message from the destination of a pending re-INVITE. This is basically the same solution as the 3rd one for MIPv6 (the preferred solution). Hence, **THE FIGURE** also applies here, with straightforward replacements, e.g., to replace B's Home Agent with B's home network SIP server, and binding update with re-INVITE. A difference is that in order to get the SIP server to be in the signalling path for the re-INVITE, the Record-Route field can be used (no modifications are needed on the mobile nodes, since SIP conveniently already has the Record-Route feature, unlike with MIPv6, where they have to be slightly modified to send control signalling to the home address of the other node rather than directly to the care-of address). The conversion of the SIP server to a pro-active one is in some ways easier than the conversion of a MIPv6 Home Agent to a pro-active forwarding proxy. This is because there is already the notion of stateful SIP servers that can retransmit messages like the re-INVITE if no acknowledgement has been received by the time a timer expires.

Sender-side and receiver-side mechanisms.   If the home network SIP server is modified to become a pro-active redirecting location proxy, instead of a pro-active forwarding location proxy, then it needs to interact with a pro-active forwarding proxy closer to the sender in the signalling path. In particular, for the case that there is a SIP server in each mobile node's home network, there would need to be a pro-active forwarding proxy in the sender's home network. This is similar to the chosen solution for MIP-LR, where the two HLRs were involved in this way. One difference is that the Record-Route feature would be needed to keep both SIP servers in the signalling path.

Evaluation.   It would appear that either the 2nd or 3rd solution is equally simple to implement, given that SIP servers of both types (forwarding and redirecting) are available. With MIPv6, on the other hand, the clear preference was for the receiver-side solution, given that Home Agents are forwarding location proxies only.

6   Discussion and Conclusions

Although the original MIP did not suffer from the simultaneous mobility problem, newer mobility management protocols like MIP-LR, SIPMM and MIPv6 do face this problem. In this paper, we have identified the problem of simultaneous mobility, introduced a new analytical framework, and then used the framework to prove some new theorems, analyse solution mechanisms, and propose and compare solutions for simultaneous mobility for MIPv6, SIPMM and MIP-LR. We have also conducted a probabilistic analysis on the likelihood of simultaneous mobility occurring. The problem is further compounded by the expected rise in popularity of at least two of the three protocols we considered, namely MIPv6 and SIPMM. Additionally, with the rise of smaller pico-cells in certain segments of the wireless market and higher mobility rates, there may be more frequent occurrences of simultaneous mobility in the future.

We explored a number of approaches to dealing with the simultaneous mobility problem. In some of the protocols, there is existing functionality that partially helps fight the simultaneous mobility problem, or that can be modified to handle simultaneous mobility. For example, with SIPMM, an RTP translator that currently only forwards data packets from the previous network to the new network can be augmented to also forward signalling, including binding updates, that might have been sent to the previous network.

References

[1] Shelley Zhuang, Kevin Lai, Ion Stoica, Randy Katz, and Scott Shenker. Host mobility using an internet indirection infrastructure. In *International Conference on Mobile Systems, Applications and Services (Mobisys)*, San Francisco, May 2003.

[2] R. Jain, T. Raleigh, D. Yang, L.F. Chang, C. Graff, M. Bereschinsky, and M. Patel. Mobile Internet access: Enhancing performance, interoperability and survivability. In *IEEE INFOCOM*, March 1999.

[3] R. Jain, T. Raleigh, C. Graff, and M. Bereschinsky. Mobile Internet access and QoS guarantees using Mobile IP and RSVP with Location Registers. In *IEEE International Conference on Communications (ICC)*, pages 1690–1695, June 1998.

[4] H. Schulzrinne and E. Wedlund. Application-layer mobility using SIP. *ACM Mobile Computing and Communications Review*, 4(3):47–57, July 2000.

[5] C. Perkins et al. IP mobility support for IPv4. IETF RFC 3344, August 2002.

[6] K.D. Wong, A. Dutta, J. Burns, R. Jain, K. Young, and H. Schulzrinne. A multilayered mobility management scheme for auto-configured wireless IP networks. *IEEE Wireless Communications Magazine*, 10(5):62–69, October 2003.

[7] D. Johnson, C. Perkins, and J. Arkko. Mobility support in IPv6. IETF RFC 3775, June 2004.

[8] C. Perkins and D.B. Johnson. Route optimization in Mobile IP. IETF work in progress *draft-ietf-mobileip-optim-11.txt*, September 2001.

[9] S. Tilak and N.B. Abu-Ghazaleh. A concurrent migration extension to an end-to-end host mobility architecture. *ACM Mobile Computing and Communications Review*, 5(3):26–31, July 2001.

[10] A. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *ACM MOBICOM*, pages 155–166, August 2000.

[11] T. Dreibholz, A. Jungmaier, and M. Tüxen. A new scheme for IP-based Internet-mobility. In *The 28th Annual IEEE Conference on Local Computer Networks*, Königswinter, Germany, November 2003.

[12] K.D. Wong, A. Dutta, K. Young, and H. Schulzrinne. Managing simultaneous mobility of IP hosts. In *IEEE Milcom*, volume 2, pages 785–790, October 2003.

[13] K.D. Wong and A. Dutta. Simultaneous mobility in MIPv6. In *IEEE Electro/Information Technology Conference (EIT)*, Lincoln, Nebraska, May 2005.

[14] 3GPP TS 23.060 v 4.9.0 (2003-12). General Packet Radio Service (GPRS); service description; stage 2 (release 4), December 2003.

[15] K.D. Wong and T.J. Lim. Soft handoffs in CDMA mobile systems. *IEEE Personal Communications Magazine*, pages 6–17, December 1997.

[16] J. Rosen, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. IETF RFC 3261, Jun 2002.

[17] R. Kravets, C. Carter, and L. Magalhães. A cooperative approach to user mobility. *ACM Computer Communications Review*, 31(5):57–69, October 2001.

[18] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobsen. RTP: A Transport Protocol for Real-Time Applications. IETF RFC 3550, July 2003.