Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/ULANC/DS/4.1.1/A1**
Contractual Date of Delivery to the CEC:	30th June 2005
Actual Date of Delivery to the CEC:	21st June 2005
Title of Deliverable:	Mobile IPv6 Handovers: Performance Analysis and Evaluation
Work package contributing to Deliverable:	WP4
Type of Deliverable*:	R
Deliverable Security Class**:	PU
Editors:	Martin Dunmore
Contributors:	Martin Dunmore, Theo Pagtzis
Reviewers:	Chris Edwards, Martin Dunmore

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Abstract:

This deliverable provides a detailed analysis of handovers in Mobile IPv6 and highlights the factors that cause unacceptable delays to certain types of applications. We show how features such as router discovery, duplicate address detection and the registration of new addresses are too inefficient to allow for seamless handovers in Mobile IPv6.

Version	Comments
1.0	First version
1.1	Fixed inconsistent usage of ‘handoff’ and ‘handover’. Fixed erroneous diagram showing the MIPv6 handover procedure.
1.2	Updated MIPv6 tests and the FMIPv6 section. Updated conclusions,

Keywords: Mobile IPv6, MIPv6, handoff, handover, performance, smooth, fast, seamless

Executive Summary

Activity 4.1 of workpackage 4 in 6NET is concerned with the support of mobility in IPv6. The Mobile IPv6 specification [1] enables hosts to change their point of attachment to the Internet whilst not breaking existing application sessions. Deliverable 4.1.1 describes the Mobile IPv6 protocol in more detail and examines a number of available implementations.

Although Mobile IPv6 solves the mobility problem in IPv6, the protocol falls short with respect to supporting applications that are sensitive to significant packet latency, or loss. This is because the procedure that is put into operation when a Mobile Node moves between networks (known as handover or handoff) can often take several seconds. This latency means that when a Mobile Node disconnects from one network it loses communication with its Correspondent Nodes until it successfully connects to another network. During this time packets destined to the Mobile Node will be lost as the Mobile Node is unreachable and the Mobile Node will not be able to send any packets.

This deliverable provides a detailed analysis of handovers in Mobile IPv6 and highlights the factors that cause unacceptable delays to certain types of applications. We show how features such as router discovery, duplicate address detection and the registration of new addresses are too inefficient to allow for seamless handovers in Mobile IPv6.

Furthermore, we show other features such as authentication and access control that are desirable in most deployed networks present a major barrier to Mobile IPv6 handover efficiency.

This deliverable is split into two parts. The first part provides an analysis of the MIPv6 handover procedure and some handover test results. The second part provides an overview and analysis of the Fast Handovers for Mobile IPv6 (FMIPv6) protocol.

Unfortunately, at the time of writing we do not know of any available implementation (compatible with a RFC 3775 compliant MIPv6 implementation) for FMIPv6 that we can test. It was hoped that the FMIPv6 implementation as part of the DAIDALOS project may have been made available for some collaborative tests. However, the implementation has not yet passed the integration testing of individual modules.

Table of Contents

1 Introduction

The Mobile IPv6 specification [1] enables hosts to change their point of attachment to the Internet whilst not breaking existing application sessions. This is achieved primarily through the Mobile Node (MN) always being reachable at its home address (HoA) via its home agent (HA). The resulting inefficiency of triangular routing can be eliminated in Mobile IPv6 with the use of route optimisation. In route optimisation, the MN issues a Binding Update (BU) to its Correspondent Node (CN) to inform the CN of its current location i.e. its Care of Address (CoA). Once the BU has been acknowledged by the CN, communication can continue on a direct path rather than a triangular one. An explanation of the general workings of Mobile IPv6 is out of scope for this deliverable. The reader is referred to [1] and Deliverable D4.1.1 for further information.

When a MN changes its point of attachment to the network, it moves from one network to another new network. This process is known as *handoff* or *handover*. During this process, the MN usually has disconnected from the old network before connecting to the new network (especially if using a single interface) and thus there is a time when the MN has lost connectivity to the Internet. During this period it cannot send or receive IPv6 packets to the detriment of existing application sessions. While many TCP applications are designed to cope with intermittent loss of connectivity by retransmitting unacknowledged packets, UDP applications will not be able to recover such losses. Furthermore, both TCP and UDP applications that rely on timely packet delivery within certain acceptable thresholds (e.g. VoIP and audio/video streaming applications) will be sensitive to the length of time a MN loses connectivity while performing handover.

Such applications desire what is know as *seamless* handovers. Where seamless refers to handovers that are both:

1. Smooth: no (or very little) packet loss
2. Fast: low latency

Thus, if the mobile Internet is to support these demanding applications, performing handovers in MIPv6 must display these two qualities. If it does not, then additional optimisations and/or changes to the protocol will be deemed necessary. It is the purpose of this deliverable to determine if MIPv6 handovers are indeed capable of supporting seamless handovers and, if not, suggest suitable improvements to the protocol architecture.

The rest of this document is structured as follows. The following section introduces the concept of handovers and describes the different types and classifications. Section three describes the Mobile IPv6 handover process in detail, from initial movement to a new network to restoration of all connections prior to movement. Section four provides an evaluation and analysis of the MIPv6 handover procedure with some test results from handover experiments. Section five looks at the Fast Handovers for MIPv6 (FMIPv6) protocol and also provides an analysis compared to normal MIPv6 handovers. Finally, conclusions are drawn in section six.

There are numerous abbreviations and acronyms used in the area of mobile and wireless networking, many of which are used frequently in this document. Consequently, the text is often difficult to follow if one is not aware of the meaning behind the acronyms. A glossary of abbreviations and acronyms is therefore provided at the end of the document.

2 Types of Handovers

The term handoff or handover[1] refers to the process of a mobile node (MN) moving from one point of attachment to the Internet to a different point of attachment. There are different types of handover according to which layers of the communication stack are affected. In general, handovers that only affect the link layer (i.e. L2) without resulting in a change of IP (i.e. L3) state are known as *horizontal* handovers. An example of this is when a MN moves between different Wireless LAN Access Points that are served by the same IP Access Router. In 802.11 terminology, both Access Points belong to the same Extended Service Set (ESS). Handovers that affect both L2 and L3 (i.e. a new IP address is obtained by the MN) are known as *vertical* handovers.

Some literature makes a distinction between *hard* and *soft* handovers. A hard handover is when all the links (usually radio) in the MN are disconnected before the new link(s) are established. Conversely, a soft handover refers to the case where the MN is always connected to the network via at least one link. In this way, there is an overlap of different link usage during the handover process. Of course, this implies either multiple interfaces or multiple radio modules on a single interface are available on the MN.

All the above types of handover may be either inter-technology or intra-technology handovers. In inter-technology handovers the handover is between different network technologies, which would usually mean separate interfaces on the MN. Intra-technology handovers are handovers of the same network technologies. Horizontal handovers would usually be of the intra-technology type, although, technically, different network technologies could be used provided the IP layer sees no change its connectivity and associated state. Vertical handovers can just as easily be inter-technology as intra-technology.

To be somewhat pedantic, one could also categorise L1 handovers such as when a Wireless LAN station switches between different frequencies and/or coding schemes of its current link. However, these are not considered to be of much relevance in the scope of this research.

2.1 Horizontal Handovers

Figure 1 shows the relatively simple case of a MN moving between APs (Access Points) and thus changing its point of attachment at the link layer. In this case, the different APs are served by the same AR (Access Router) and will most likely belong to the same 802.11 *Extended Service Set* (ESS). The MN is still considered to be attached to the same ‘link’ from the point of view of the IPv6 layer. Thus, the MIPv6 handover procedure is not triggered because the MN can still use its current CoA (Care of Address). In fact, the IPv6 layer should be completely unaware that movement between APs has taken place provided that the inter-AP handover does not disrupt any IPv6 communication.

[1] Both ‘handoff’ and ‘handover’ are used frequently in literature although they have the same meaning.

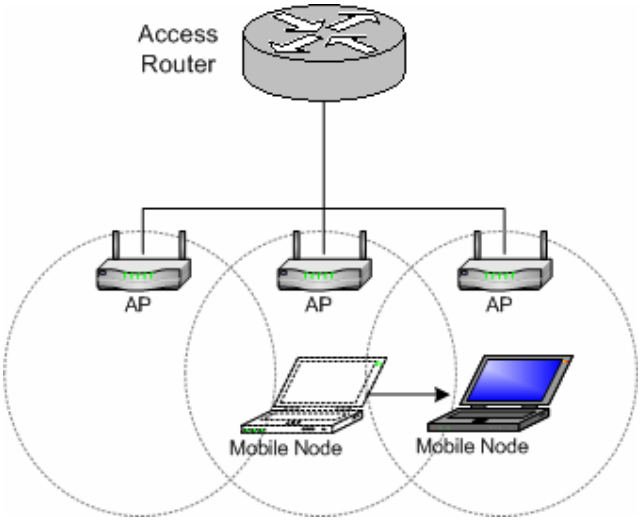

Figure 1 Moving within same ESS

As the MN roams between AP coverage areas, known as a Basic Service Set (BSS) in 802.11 terminology, the 802.11 client in the MN continually monitors the signal strength of all the different BSSs within the same ESS. It is this signal strength information that is used to decide if the station (i.e. the MN) should perform a handover between APs.

2.2 Vertical Handovers

Figures 2, 3 and 4 show examples of a MN performing vertical handovers.

In the example of Figure 2, the MN moves between APs that belong to different ESSs and which are served by different ARs. This means that the MN would no longer be reachable from its previous AR and must now use its new AR.

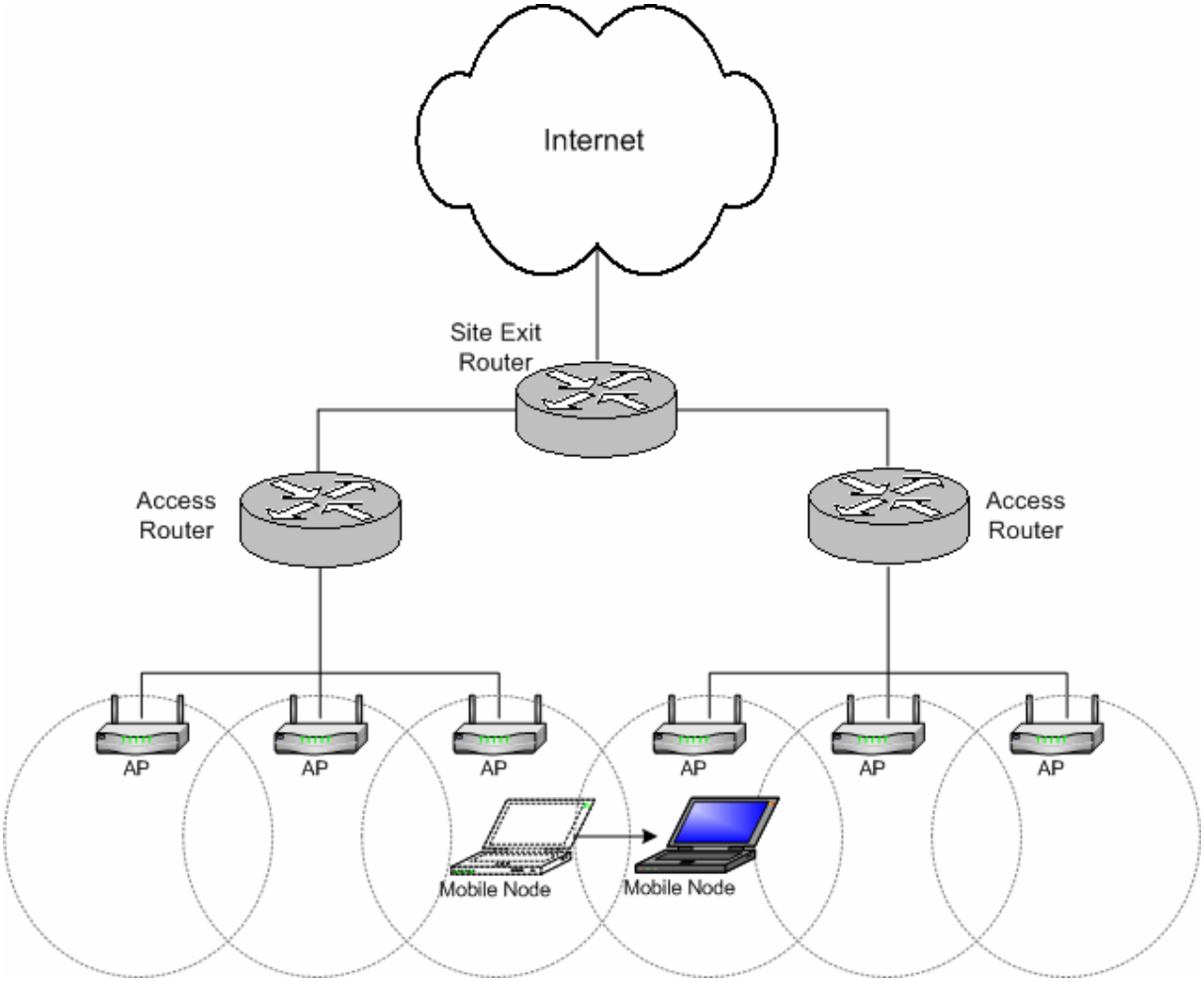

Figure 2 Moving between ARs of the same provider

However, in this example, both the PAR (Previous Access Router) and the NAR (New Access Router) belong to the same administrative domain (i.e. service provider). Although the MIPv6 handover procedure must be activated, it is likely that any 'higher layer' state such as AAA or QoS information would not need to be re-negotiated within the same provider during handover, thus lessening the overall handover latency somewhat.

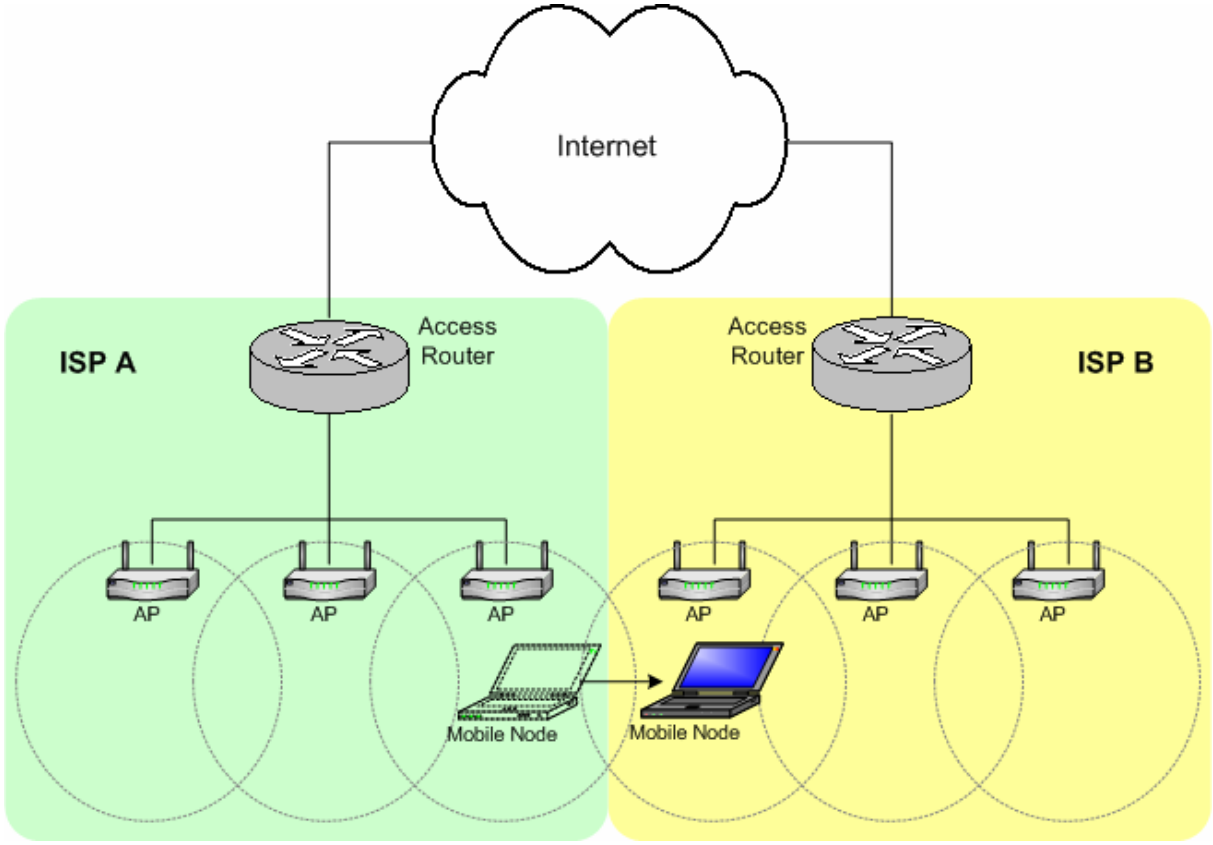

Figure 3 Moving between ARs of different providers

In the example of Figure 3, the MN has changed provider networks. Although geographically close, roaming to another AP owned by a different provider may well result in the MN moving a great distance topologically speaking. A good example of this is a MN moving between WLAN hotspots of different providers that are physically close to one another (e.g. between Starbucks and McDonalds). Another example would be a MN moving out of range of any WLAN coverage and thus 'falling back' to a 3G network employing GPRS or UMTS. Typically, the providers of the WLAN coverage and the 3G network coverage will be different. This scenario is illustrated below in Figure 4.

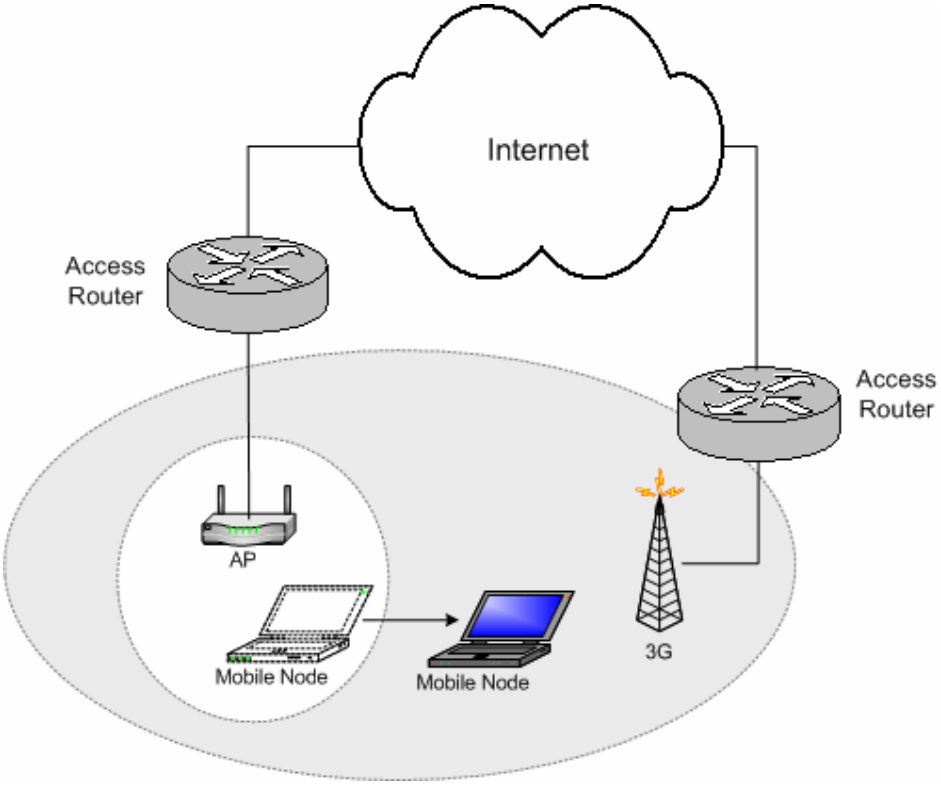

Figure 4 Moving between WLAN and 3G networks of different providers

In all of the above examples the MIPv6 handover procedure will be activated. Typically, the latency involved in the handover procedure would be greater in the case of vertical handover between different provider networks. This is not only due to the fact of the topological distance between the PAR and NAR is generally greater. Note that many providers will block all outbound traffic from the MN until authentication and authorisation have been satisfied. Thus, MIPv6 handover success in this scenario depends on successful authentication and authorisation on the new network by whichever methods are employed to accomplish this. The following section examines the MIPv6 handover procedure in detail.

3 The Mobile IPv6 Handover Process

The Mobile IPv6 (MIPv6) specification [1] is a proposed standard by the IETF to provide transparent host mobility within IPv6. The protocol enables a Mobile Node (MN) to move from one network to another without the need to change its IPv6 address. A Mobile Node is always addressable by its *home address*, which is the IPv6 address that is assigned to the node within its home network. When a MN is away from its home network, packets can still be routed to it using the MN's home address. In this way, the movement of a node between networks is completely invisible to transport and other higher-layer protocols.

When a MN changes its point of attachment to the Internet from one IPv6 network to another IPv6 network (also referred to as *roaming*), it will perform the MIPv6 handover procedure. The MIPv6 handover procedure is similar to the autoconfiguration procedure an IPv6 node booting up onto a network but has some important differences:

- the MN must somehow detect that it has moved onto a new network.
- once configured, the MN must inform its home agent (HA) and each correspondent node (CN) of its new location.
- during the handover procedure, upper layer connections will still be active so the handover procedure should be performed as quick as possible to minimise disruption from lost and severely delayed packets.

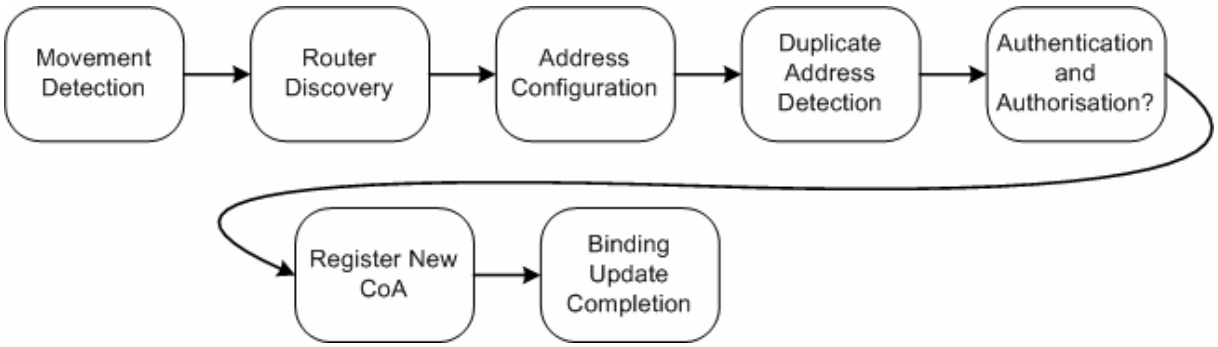

Figure 5 The MIPv6 handover procedure

The MIPv6 handover procedure is illustrated in Figure 5 and is described in more detail in the following sections. In current IPv6 specifications, each stage of the procedure is mandatory with the exception of authentication and authorisation, although this stage will be present, at least in some form, in most deployed networks.

3.1 Movement Detection

In Mobile IPv6, it is generally the responsibility of the MN to detect that it has moved between networks. Determining whether or not a MN has moved networks is not always a simple issue. However, the general rule of thumb that a MN has moved can be seen as:

1. the current access router is no longer reachable, and
2. a new and different access router is available

In order to determine if its CAR (Current Access Router) is still bi-directionally reachable the MN performs *Neighbour Unreachability Detection* on a continual basis.

Neighbour Unreachability Detection works in the following manner. When an IPv6 host has a packet to send, it checks the Neighbour Cache to determine the link layer address of the next hop node (either an on-link neighbour or a router). The Neighbour cache also has an associated state with each neighbour entry. A neighbour state of REACHABLE indicates that the neighbour is considered reachable.

In IPv6 a host considers a neighbour reachable if it has recently received confirmation that packets sent to the neighbour have been received. This is achieved in two ways: the receipt of a neighbour advertisement from the neighbour in response to a neighbour solicitation sent by the host, or a hint from upper layer protocols. The IPv6 stack utilises the acknowledgements of upper layer protocols to register the fact that a packet has recently been received from a given destination address and so is considered reachable.

The IPv6 host will send a neighbour solicitation in the event that the neighbour cache entry not being set as REACHABLE when there is a packet to send[1].

Note that neighbour unreachability detection only occurs when the MN has a packet to send. Thus, in the worse case scenario when the MN is not any sending packets, it may not notice that it has moved networks until it receives an unsolicited router advertisement from the new on-link router (consistent with the normal router advertisement interval). Unfortunately, this may be the case when the MN is receiving real-time streams when an interruption in connectivity can cause packet losses and unacceptable latency while the new handover is taking place. In such a scenario, the MN may not actually be transmitting much data itself, perhaps occasional TCP or application layer acknowledgements, but nothing that will allow the unreachability of its CAR (Current Access Router) to be discovered in a timely fashion.

However, the MN noticing a new router advertisement only serves as a *hint* that the MN has moved networks and does not guarantee it. For example, one possibility could be that a new (additional) router has been activated on the existing link. Furthermore, as stated in RFC 2461 [3] unsolicited router advertisements must not be used as confirmation of bi-directional router reachability since they only confirm reachability in the router to MN direction.

3.2 Router Discovery

Router Discovery is achieved through the receipt of a router advertisement sent from the *New Access Router* (NAR). This will either be in the form of a router advertisement sent periodically to the all nodes multicast address, or in response to a router solicitation sent by the MN. There is a potential race condition here. The MN will send a router solicitation if it discovers that its CAR is considered unreachable (i.e. its neighbour cache entry is not set to REACHABLE), and will thus receive a solicited router advertisement from the NAR, or it will receive an unsolicited router

[1] This may involve a wait of DELAY_FIRST_PROBE_TIME seconds if the neighbour cache entry is in the DELAY state

advertisement from the NAR as part of its periodic broadcasts. There is no guarantee as to which method will occur first. It will depend on the exact circumstances at the time of handover: the period or router advertisement transmissions by the NAR and the exact value of the various timers at that moment in time. One can hypothesise that reducing the period of router advertisements will increase the likelihood of receiving an unsolicited router advertisement on the new link before realising that the PAR is no longer reachable.

However, as noted earlier the receipt of a new unsolicited router advertisement is not necessarily a definite indication of having moved networks. Thus the MN may also decide to confirm that its CAR is definitely unreachable before deciding to use the NAR. This would involve transmitting neighbour solicitations for a pre-determined time without receiving a corresponding neighbour advertisement.

3.3 Care of Address Configuration

The MN must configure itself with an IPv6 address to be used on the new network. This will be the MN's *New Care-of Address* (NCoA). Address configuration can be performed in a stateful or a stateless manner. An IPv6 host may use both stateless and stateful address configuration completely independently from one another. The precise method to be used can be signalled with the setting of various flags in router advertisement messages.

Of course, a host may also be configured by manual means. Needless to say, any address configuration requiring manual input from the user would be a catastrophe for an expedient MIPv6 handover. However, most sane network operators would not allow or even wish for its users to manually configure addresses. However, it is possible that state other than IPv6 addresses may be left to the user to configure manually.

3.3.1 Stateless Address Configuration

There are two ways in which an IPv6 node can configure its address in a stateless fashion:

1. Using automatic address configuration with prefix discovery
2. Using stateless DHCPv6

Automatic address configuration utilising prefix discovery is specified in [4]. If the 'autonomous' flag of a Prefix Information Option contained in a router advertisement is set, the IPv6 host may automatically generate its global IPv6 address by appending its 64-bit interface identifier to the prefix contained in the router advertisement. There are different ways in which the host may choose how to generate its interface identifier (e.g. based on MAC address, random or cryptographically generated). However, this is out of scope for this document.

Stateless DHCPv6 is not mentioned as an option given in router advertisements [3]. However recent discussions in the IPv6 w.g. have suggested signalling the usage of stateless DHCPv6 via the 'O' flag in router advertisements. At the time of writing the exact way of signalling that hosts should use stateless DHCPv6 is not clear. However, since there are few available implementations, this is not a major concern.

If DHCPv6 (stateless or stateful) is to be used by the host for address configuration it incurs an extra overhead that is detrimental to expedient handovers. DHCPv6 requires an extra request/response exchange on the new network in addition to normal router discovery mechanism.

3.3.2 Stateful Address Configuration

As far as the handover is concerned, using stateful DHCPv6 is no different to using stateless DHCPv6 as the observed request/response times should be the same in most cases. However, it is possible that the extra overhead of reading and writing state to memory inside the DHCPv6 server may lead to a small increase in latency when compared to its stateless equivalent.

Speaking purely from the perspective of a MIPv6 handover, using stateless address configuration with prefix discovery is probably the preferred option. This is simply because it incurs less latency that using DHCPv6 or any other stateful mechanism.

It is worth noting however, that most network operators (certainly those in 6NET) seem to favour the use of stateful DHCPv6 for the extra control and documenting of address assignments.

3.4 Duplicate Address Detection

Just as a node must perform DAD (Duplicate Address Detection) when it boots up onto an IPv6 network to ensure that its configured addresses are likely to be unique on the link, a MN that moves onto a new network must perform DAD on the CoA that it obtains from the CoA configuration phase. This holds true regardless of whether the CoA address has been obtained by stateless, stateful or manual means.

In IPv6, the DAD procedure is defined in RFC 2462 “IPv6 Stateless Address Autoconfiguration” [4], and uses the neighbour discovery procedures defined in RFC 2461 [3]. A MN cannot begin to use a new CoA until the DAD procedure has been successfully executed. Until DAD has succeeded, the MN’s new CoA is seen as *tentative*, in that it can only be used for neighbour discovery purposes (of which the DAD procedure is part of). If a MN was to use its new CoA before successful DAD and another node was using the same address on the link, the MN would erroneously process packets intended for the other node.

To perform DAD, the MN sends out a neighbour solicitation message with its own new CoA address as the target address of the solicitation message. The destination address in the IPv6 header of the neighbour solicitation is set to the solicited-node multicast address of the target address with the source address being the unspecified address. If there is another node on the link that is using the same address as the MN’s new CoA, one of two things will happen:

1. The duplicate node will receive the MN’s neighbour solicitation message and reply with a neighbour advertisement (sent to the all-nodes multicast address) thus exposing the duplicated address to the MN.
2. The MN will receive a neighbour solicitation with its new CoA as the target address from a duplicate node that is also in the process of performing DAD.

Thus, the DAD procedure will give an explicit indication to the MN should there be another node on the network that is using its new CoA. However, (and to the detriment of any node wishing to perform autoconfiguration at haste) the DAD procedure provides no explicit indication that a MN’s new CoA is *not* being used by another node on the network. Indeed, the point at which DAD can be considered to have succeeded is quite vague. According to RFC 2462, a node performing DAD can consider its tentative address unique if no indications of a duplicate address is observed within RETRANS_TIMER milliseconds after sending DUP_ADDR_DETECT_TRANSMITS number of neighbour solicitations. Both the values of RETRANS_TIMER and

DUP_ADDR_DETECT_TRANSMITS are configurable parameters and by default are set to 1,000 and 1 respectively. Therefore, under default conditions DAD will take a minimum of 1,000 milliseconds (1 second) plus additional delay for link transmissions and logic computation.

Note that RFC 2462 states that a node *should* delay sending its neighbour solicitation for DAD by a random time interval between 0 and MAX_RTR_SOLICITATION_DELAY seconds if it is the first packet sent from the interface after (re)initialisation[1]. In RFC 2461, MAX_RTR_SOLICITATION_DELAY is defined as being 1 second in duration. Therefore, unless the MN has previously sent a router solicitation, it will incur further delay during its autoconfiguration process if its code is simply following recommendations laid down in current IPv6 specifications. In the average case (assuming a pure random function) this will be an extra 500ms, and up to 1000ms (1 second) in the worst case.

It is arguable that a MN that frequently attaches itself to different networks, and consequently configures different CoAs on a frequent basis, runs a greater risk of picking up a duplicate address on link at some point during its lifetime (although, this should still be a small risk). The probability of duplicate addresses occurring relates to the method of address configuration used. If stateless addressing is used, the 64-bit host ID part of a node's IPv6 address is generated from the node's link-layer interface identifier (e.g. MAC address). These identifiers are highly unlikely to have duplicates although this has been known to occur due to manufacturer error or malicious assignment by users.

Note that in order to speed up the autoconfiguration process, a MN (or indeed any node) may choose to initiate DAD in parallel to router discovery. Since the value of the node's link-layer identifier is known in advance, the MN can perform DAD on its link local address before receiving a router advertisement. If the router advertisement instructs the node to use stateless address configuration, the MN need not perform DAD on its resultant global unicast address if it has already verified the uniqueness of it's link-local address.

As a router may delay responding to a router solicitation by a few seconds, a MN that performs DAD only after receiving a valid router advertisement may experience significantly longer autoconfiguration latency than performing the steps in parallel when stateless addressing is used. However, as noted above, a MN may only detect that it has moved onto a new network as a result of receiving a new router advertisement; in which case the potential speed up of performing DAD in parallel to router discovery is lost.

3.5 Configuration of Other IPv6 State

3.5.1 AAA State

If a MN moves across different administrative domains it is likely that will encounter some form of AAA infrastructure that must be negotiated before access to the new network can be granted.

In the event of AAA establishment the set of interactions involved encompass a handshake between the MN, the local AAA server (AAAL) and the MN's home AAA server (AAAH). An 'attendant'in the local network will ask for credentials from the MN and pass this on to the AAAL. The AAAL

[1] This is to avoid potential link congestion when multiple nodes configure simultaneously, such as after a power failure.

will then need to verify the identity of the MN with the MN's AAAH before it can grant access to the network. This implies that there is a requirement for at least one RTT between the AAAL and the AAAH to verify the MN and then another RTT between the AAAL and the MN to acknowledge verification. This may be reduced to a single RTT if the AAAL is co-located at the NAR. The important point is that one full RTT is required to authenticate and bill the MN. The size of this latency is dependent on the exact locations of the MN, AAAL, AAAH and the particular type of AAA implementation involved. Suffice to say, the incurred latency will be beyond that which is needed for the handover to be considered seamless.

Needless to say, a failure of AAA establishment due to lack of credentials, errors in the network, errors with authentication protocols etc., is disastrous to the handover because access to the new network will be refused.

3.5.2 QoS State

Should the MN have particular QoS requirements it may need to convey this information to the new network. Using an Integrated Services/RSVP approach, the QoS signalling occurs separate to actual data transmission and therefore should not incur any additional latency for the handover. In other words, no QoS state has to be established in order for the handover to be successful. Of course, should required QoS parameters fail to be negotiated before the handover completes, or the QoS requirements are rejected by the new network, then existing application sessions may suffer as a result.

A similar situation exists when using Differentiated Services (DS). No prior negotiation with any DS QoS broker and related policy servers need to have happened for the handover to be successful and so handover latency should not be affected. However, late successful negotiation of DS parameters or failure may harm existing application sessions that rely on certain levels of QoS in the network. One obvious example here is of using VoIP from a MIPv6-enabled WLAN device while performing a handover between networks.

3.6 Registration of New Care-of Address

Once the MN has detected that it has moved networks, obtained a new CoA and has been granted access to the network, it must inform its HA (Home Agent) of its new location. During the time from when the MN lost connectivity with its PAR until it informs its HA of its new location, all packets that have been sent to it will have been lost and it will not have been able to send packets to any of its CNs. The MN registers its NCoA with its HA by sending it a binding update (BU). The HA acknowledges this by replying with a binding acknowledgement (BA) and is then able to tunnel packets bound to the MN's home address (HoA) to the MN's new location (i.e. the MN's NCoA).

3.7 Binding Update Completion

This stage refers to the MN informing all of its CNs as to its new location and that it is reachable at its NCoA.

As with registering its new location with its HA, the MN sends a BU to each CN to inform them of its new location. However, an additional procedure is followed for BUs that are sent to CNs. The procedure is known as a 'Return Routability' (RR) test and is used as a way of satisfying the CN

that the BU it receives is authentic and not from a malicious third party. A thorough discussion of RR is out of scope for this document and the reader is referred to [1] D4.1.1.

In brief, RR uses a Home Test (HoT) and a Care-of Test (CoT). The CN issues the two tests to the MN via the HA and the route optimised path (i.e. direct to the NCoA) respectively. The MN replies with the answer to the two tests in the BU that it sends to the CN. If the tests are answered correctly, the CN acknowledges the BU.

Once the MN has received BA's from its CNs, the handover process can be considered completed. It is arguable that the handover can be considered as completed once the NCoA has been registered with the HA. However, the optimum handover will see all optimised CN sessions restored to their optimised state.

4 Analysis and Evaluation of MIPv6 Handovers

As can be seen from the previous section, the Mobile IPv6 standard relies fundamentally on core IPv6 protocol functions and in particular the base IPv6 specification [2] and IPv6 neighbour discovery [3]. Analysing the behaviour Analysing the behaviour of the MIPv6 standard over Wireless LAN we identify that the latency that can significantly affect handover delay during an IPv6 handover comprises of the following components:

- Movement detection time (t_d): this is the time required by the MN to detect and establish that it has moved to a new point of attachment (i.e. the discovery of a new on-link router).
- IP CoA configuration time (t_a): this is the time between the establishment of having moved and the time that a globally routable IPv6 address has been configured (this includes duplicate address detection).
- Context establishment time (t_c): this is the time between the establishment of a globally routable care-of IPv6 address and the establishment of the appropriate context state. Example context states are AAA state or QoS state.
- Binding registration time (t_r): past the establishment of context-specific state of the MN, this is the time between the dispatch of a binding updated signal to the HA to the receipt of an acknowledged BU piggybacked on the first packet from its communication peer.
- Route optimisation time (t_o): this is the time from registering the new CoA with the HA to completing route optimisation with the current list of CNs. This includes the return routability procedure which, if used, must occur before a BU is sent by the MN to a CN.

The total IP handover delay may thus be represented by (t_h) defined as the sum of the aforementioned latency components as follows:

$$t_h = t_d + t_a + t_c + t_{r+} t_o$$

Figure 6 illustrates a simple handover procedure without any AAA or QoS state establishment (t_c).

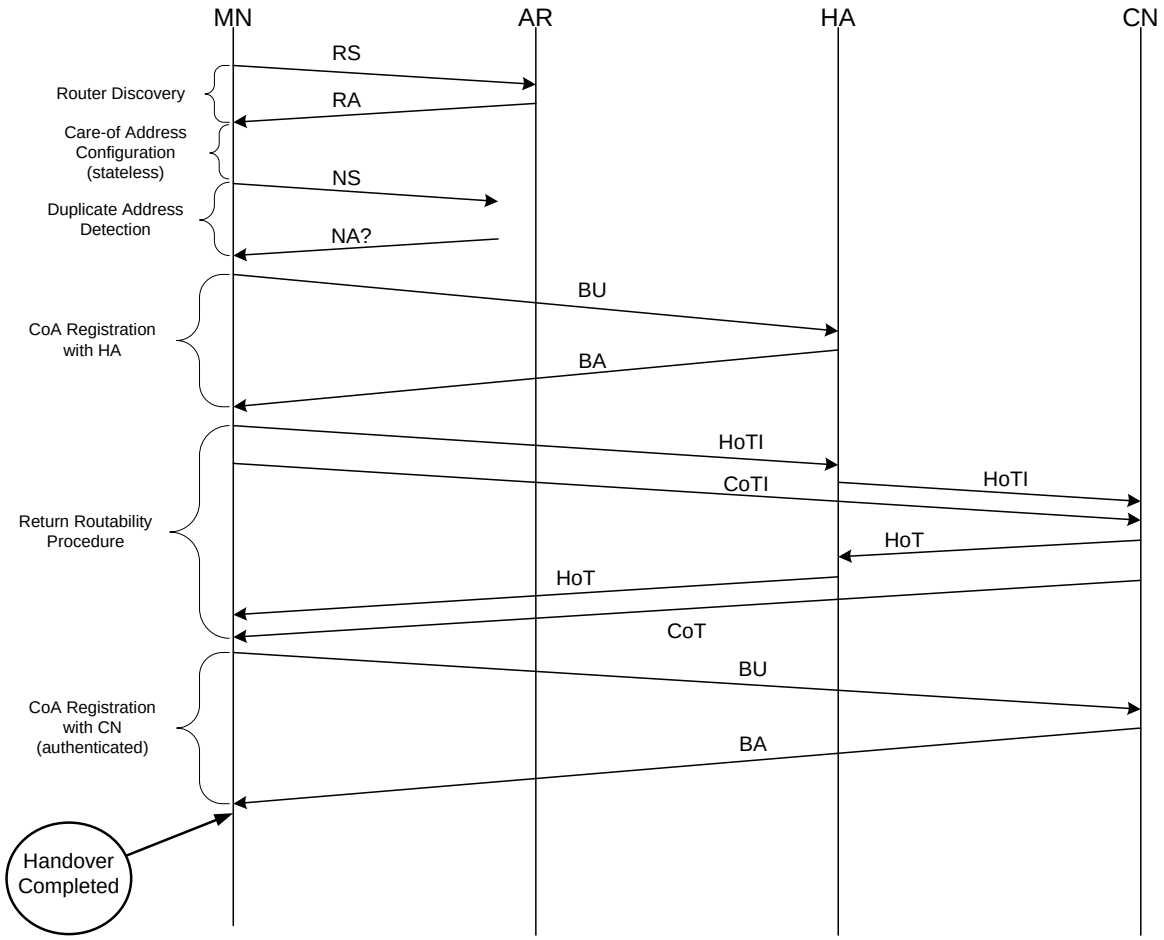

Figure 6 Handover Procedure

4.1 Movement Detection Time

The movement detection time (t_d) is the sum of two individual latency components:

1. Link switching delay (T_{l2}): this is the time delay pertaining to the re-association of the wireless station in a 802.11 wireless LAN with its Basic Service Set (BSS) Access Point (AP).
2. Link-local IPv6 address configuration delay (T_{ll}): this is the time between the first time that the MN encounters a new link by receiving neighbour adverts over its all nodes or solicited-nodes multicast address and configuration of a link-local address. Configuration of a link-local address is effected as well as L2 information is exchanged with the new AR.

The movement detection time can thus be expressed as:

$$t_d = T_{l2} + T_{ll}$$

4.2 IPv6 CoA Configuration Time

We define CoA configuration time (t_a) as the time commencing from the moment of the receipt of a router advertisement (including the router advert solicitation if so used) to the moment that Duplicate Address Detection and the update of the routing table has completed. Depending on the mechanism employed to configure an IPv6 CoA t_a may vary. In particular, for stateless IPv6 address auto-configuration [4] t_a is comprised of the following delay components:

$$t_a = T_{prefAdv} + T_{AddrConfig} + T_{DAD} + T_{RouteUpdate}$$

where $T_{prefAdv}$ is defined as:

T_{rtAdv} - T_{rtSol} (if the router advertisement is solicited)

rtAdvInterval / 2 (if router advertisement is periodic)

$T_{AddrConfig}$ is the time required by the MN to employ the address configuration rule such as EUI64, to produce a unique, globally routable IPv6 address. We anticipate that this latency component is dependent on the processor speed of the MN and as such may be negligible compared to the total of t_a.. T_{DAD} is the time required to resolve uniqueness of the configured IPv6 CoA. The mechanism to effect this is typically address resolution by transmitting a Neighbour Solicitation for this address to the all-nodes multicast address and then waiting for *RetransTimer* interval (default 1000ms) before transmitting up to *DupAddrDetectTransmits* (default 1). If during or after *RetransTimer* interval there has been no Neighbour Advertisment on the particular tentative CoA, the address is assumed to be unique and is assigned to the interface. For the purposes of comparative analysis this research assumes use of the default values in Mobile IPv6 since these are expected to be the standard default configuration values. Given time we may experiment with optimisations on these values for Mobile IPv6; this is however out of the scope of our hypothesis.

In the event of stateful address autoconfiguration (i.e. DHCPv6 [5]) the time for CoA configuration becomes:

$$T_{AddrConfig} = T_{DHCPaddrReq} + T_{DHCPaddrResp} + T_{RouteUpdate}$$

$T_{DHCPaddrReq}$ and $T_{DHCPaddrResp}$ represent the transmission delay incurred by stateful configuration of a CoA via a DHCP server.

Figure 7 illustrates the MIPv6 handover procedure where DHCPv6 is used for CoA configuration.

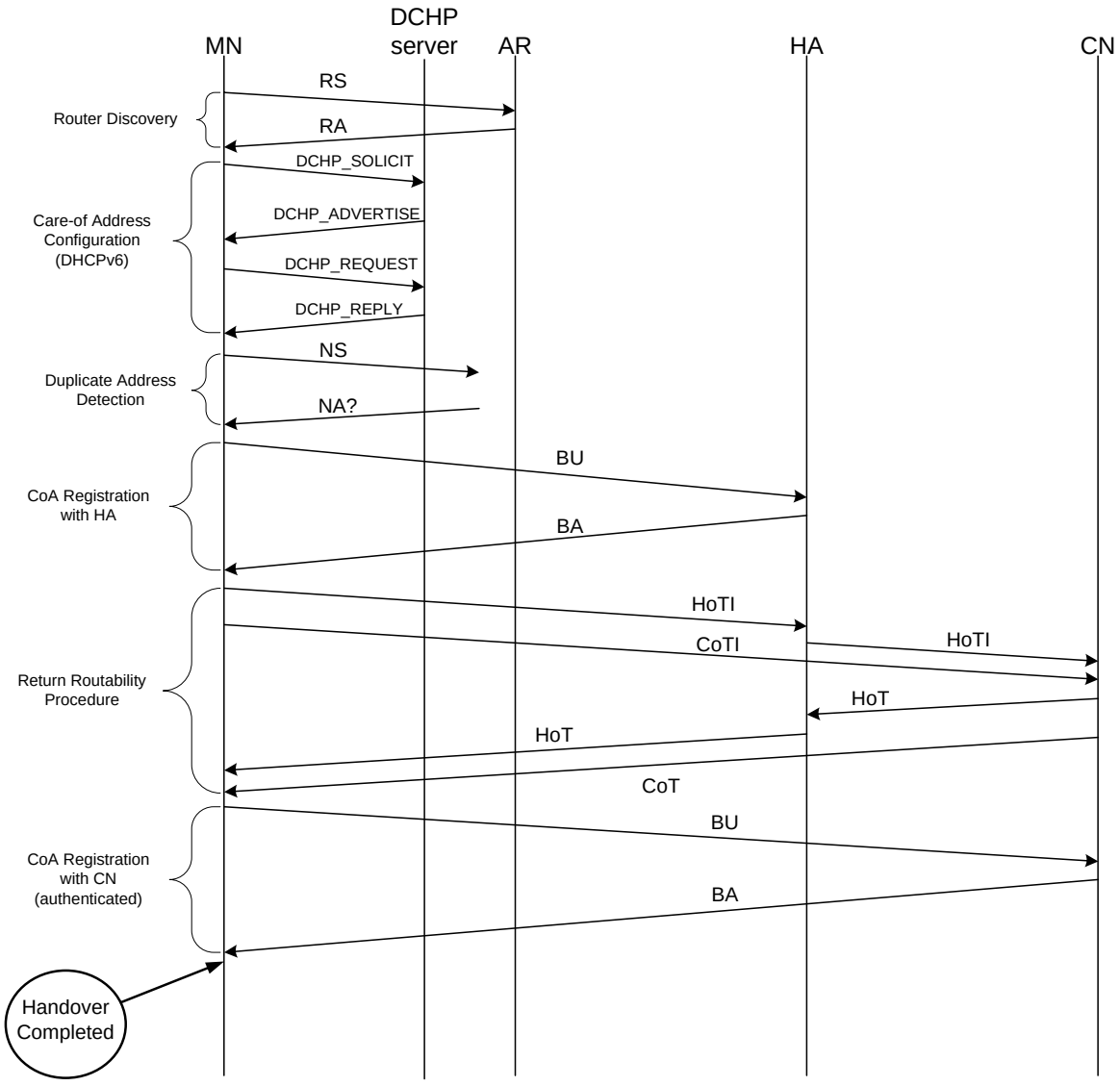

Figure 7 Handover Procedure using DHCPv6

Note it is quite likely that DHCPv6 is used even in the case of the MN using stateless address configuration as instructed by the received RA. For example, a list of local DNS or NTP servers can be provided by the DHCPv6 server. In the case when DHCPv6 is not used for address configuration, using it for additional information should not affect the handover latency assuming it is done in an asynchronous manner.

At this point it is worth making a comment about Duplicate Address Detection (DAD). It seems at first glance that constantly performing checks for address duplication for a very uncommon event is not a wise course of action. Consider that DAD is performed for every node, every time it joins a network (either through booting up or moving into it as with MIPv6). Thus, we are performing DAD 100% of the time. Obviously, DAD is critical to the smooth operation of IPv6 as duplicate addresses, resulting from either malicious or erroneous configuration can render the link unusable

for the associated nodes. However, it seems somewhat of an overkill to perform DAD 100% of the time in order to catch an event that, theoretically speaking, has only a 1 in 2^{64} chance of occurring[1]. As a consequence of this, some thought has been given to providing an 'optimistic' algorithm for DAD [9]. In the optimistic DAD algorithm RETRANS_TIMER delay is removed so that the node can begin using the address immediately after generating an address (by stateless or stateful means). The node also transmits a Neighbour Advertisement (with the override bit set to zero) in order to create entries in the Neighbour Caches of other nodes on the link.

It is hoped that WP4 will be able to perform handover tests using optimistic DAD later in the project, subject to available implementations of the algorithm.

4.3 CoA Registration Time

The CoA registration time (t_r) is defined as the transmission delay incurred during registration of the MN CoA with its HA. This is essentially the RTT between the MN and HA plus associated processing of the BU and BA messages.ack bindings with its longest-RTT peer in two distinct cases depending on the mode of security effected in the BU registration process:

$$t_r = RTT_{MN\text{-}HA} + BU_{proc} + BA_{proc}$$

4.4 Route Optimisation Time

The route optimisation time (t_o) is defined as the transmission delay incurred during registration of the MN bindings with the CN that is furthest away (i.e. with the longest RTT) in two distinct cases depending on the mode of security effected in the BU registration process:

$$t_o = RTT_{MN\text{-}CN} + BU_{proc} + BA_{proc} \text{ (if BU is not authenticated)}$$

$$t_o = T_{HoT\text{-}CoT} + (RTT_{MN\text{-}CN} + BU_{proc} + BA_{proc}) \text{ (if BU authenticated)}$$

In the event of an unauthenticated BU, the route optimisation time t_o is defined as the time period between a BU dispatched to the CN and the first data packet received by the MN from the CN. The BA from the CN is typically piggybacked in the first data packet.

In the event of an authenticated using return routability (RR). The MN must first initiate the Home Test (HoT) and Care-of Test (CoT) before it can send a binding update. The RR procedure is illustrated in both Figure 6 and Figure 7.

Note that once the HA has acknowledged the registration of the MN's new CoA, any *new* CN attempting communication with the MN will succeed due to the HA being able to tunnel packets destined for the MN to its new CoA (i.e. using triangular, non-optimised routing). However, communication with any existing CN at the time the handover occurred and with whom route optimisation was being used, cannot resume until the MN has successfully registered its new CoA with it by performing the route optimisation procedure.

[1] Real probabilities of collision will be less than this theoretical maximum, but will still be extremely rare events.

4.5 Early Test Results

This experiment measures the time taken for a vertical handover, using the event driven movement detection procedure for Linux PCMCIA card services. Both the Mobile IPv6 stack and the PCMCIA services were profiled to log the system time at strategic points during the configuration, thus allowing the most heavyweight procedures to be identified Figure 8 shows a timeline of an average vertical handover. This timeline was generated under a situation we call 'cold handover', i.e. where the network device is initially totally unconfigured. Note that the majority of the time is spent initialising the device driver, before any network configuration can take place.

Once the device driver is running, it still takes time for the interface to become operational. This is due to the architecture of the PCMCIA card services, which has a user level component to enable the interface. Once the interface is enabled, the IPv6 stack is notified and stateless address autoconfiguration begins to find the mobile node's new care-of address. Around 160ms later, an address has been acquired, and binding update messages are transmitted from the node.

On our local area testbed where the RTT are generally small, binding acknowledgements are received within 5ms. This makes a total latency of 650ms, from card insertion to complete network access. This latency drops to 165ms under 'warm handover', where the device is already configured at the link layer, and only dynamic address autoconfiguration and binding update transmission is required. Handover times of approaching 5ms can be achieved during 'hot handover', whereby multiple interfaces are run in parallel, and care-of addresses can be acquired before the handover takes place.

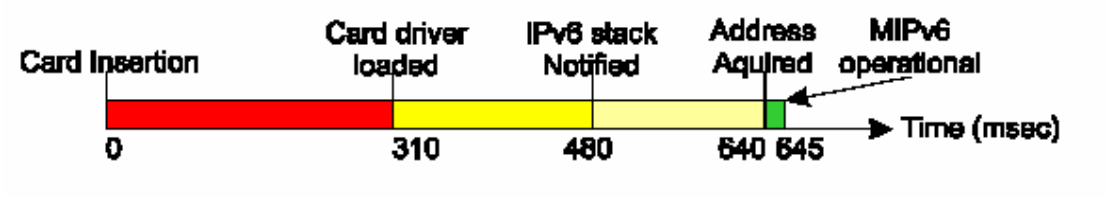

Figure 8 Mobile IPv6 Handover Overhead

To complete the test suite, the MIPv6 stack was tested against a real multimedia application – an IPv6 capable Video on Demand (VoD) system.. This application provides high quality video streams over an IPv6 infrastructure. A 1.5Mbps MPEG1 video clip was streamed to the home address of the mobile node, which was connected to its home subnet via Ethernet. A cold handover was then performed to a 2 Mbps wireless LAN, connected to a different IPv6 subnet. The route optimised handover completed with a relatively short (0.5 – 1.5 sec) break in audio, and associated stalling of the video for a similar time. A similar phenomenon was observed on a handover back to the home subnet.

These delay figures suggest that allocation of IPv6 addressing state during an IP handover, generates by itself enough transmission delay (~160ms) to place any active IPv6 flows on the boundaries of acceptable guarantees for real-time traffic delivery. The experiment involved a rather old P133 laptop equipped with 10Mbps Ethernet PCMCIA interfaces running over Mobile IPv6. No wireless LAN connection was involved in this experiment. While the laptop used in this experiment is surpassed by today's mobile computing devices by an order of magnitude, the results bear some

relevance to the initial claim of delay incurred during IP addressing state establishment for two reasons:

1. Truly mobile devices such as today's PDAs are not expected to be significantly powerful given the stringent energy requirements of their embedded hardware (colour LCDs, wireless interfaces) and their cost/performance ratio.
2. The delay observed is manifested over Ethernet interfaces. This implies, no signalling latencies due to RF modulation or propagation effects. An Ethernet PCMCIA interface can detect colliding packets at its MAC sublayer as its can implement full duplex transmissions. A Wireless PCMICA interface of any signalling rate imposes increased contention due to the inability of achieving full duplex communications, while its MAC layer effects collision avoidance through contention-based coordination techniques. That is to say, the Ethernet MIPv6 handover experiments have removed any effects introduced by the wireless MAC sublayer, on IP handover delay. This is especially the case for DAD resolution over the air interface.

The above figures do not represent an unrealistic assumption and starting point for the purposes of this research given the fact that High Rate (HR) 802.11b/g employ dynamic rate shifting toward the perimeter of the coverage area; that is to say, for WLAN-connectivity provisioned by real-world WISPs, the MN is expected to encounter significantly lower signalling rates (1-2 MBps) on the edges of the coverage area of its current point of attachment, where an IP handover is imminent. While we employ these figures as an initial starting point for the purposes of our simulations, during the course of this research we further confirm the validity of these assumptions by repeating these IPv6 handover experiments over 802.11b WLAN interfaces as well as traditional Ethernet.

4.6 Further tests

Figure 9 shows the small MIPv6 testbed used for performing the handover tests. A MN running MIPL v1.1 is away from home (the HA also running MIPL v1.1) and can attach to one of two networks represented by the SSIDs ‘roam1’ and ‘roam2’. We decided to conduct handover tests from one foreign network to another (e.g. rather than from home network to foreign network) as the nature of mobility implies that when one is mobile, one is very rarely located at the home network.

Handovers from one network to another were forced by turning off one of the APs so that the MN would immediately associate with the other AP and thus receive different RAs than on the previously connected network.

The handover times were measured from the point at which the AP is switched off (link down notification) to when the Binding Acknowledgement is received from the Correspondent Node with each event being timestamped in the relative logs. Note that the Return Routability protocol was enabled for Route Optimisation.

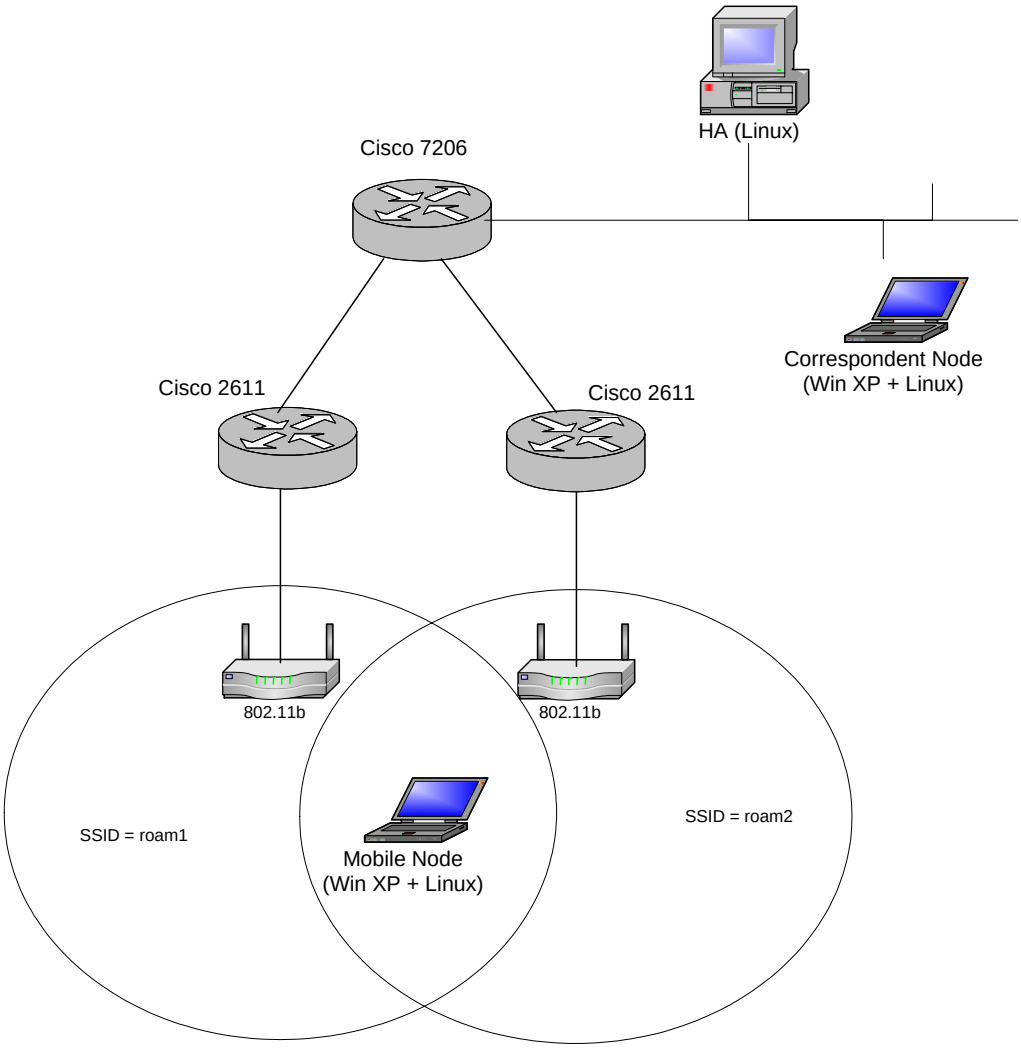

Figure 9 Simple MIPv6 Handover Testbed

4.6.1 Reducing the Router Advertisement Intervals

In order to demonstrate the effects of reducing the RA interval we performed handover tests with various configurations of RA intervals on the Cisco 2611 access routers. As described earlier, upon detecting movement, the MN will issue a RS assuming it hasn't received a new RA already. As can be seen from Figure 10, the time it takes for the MN to receive a solicited RA is fairly random within a given (configurable) time window.

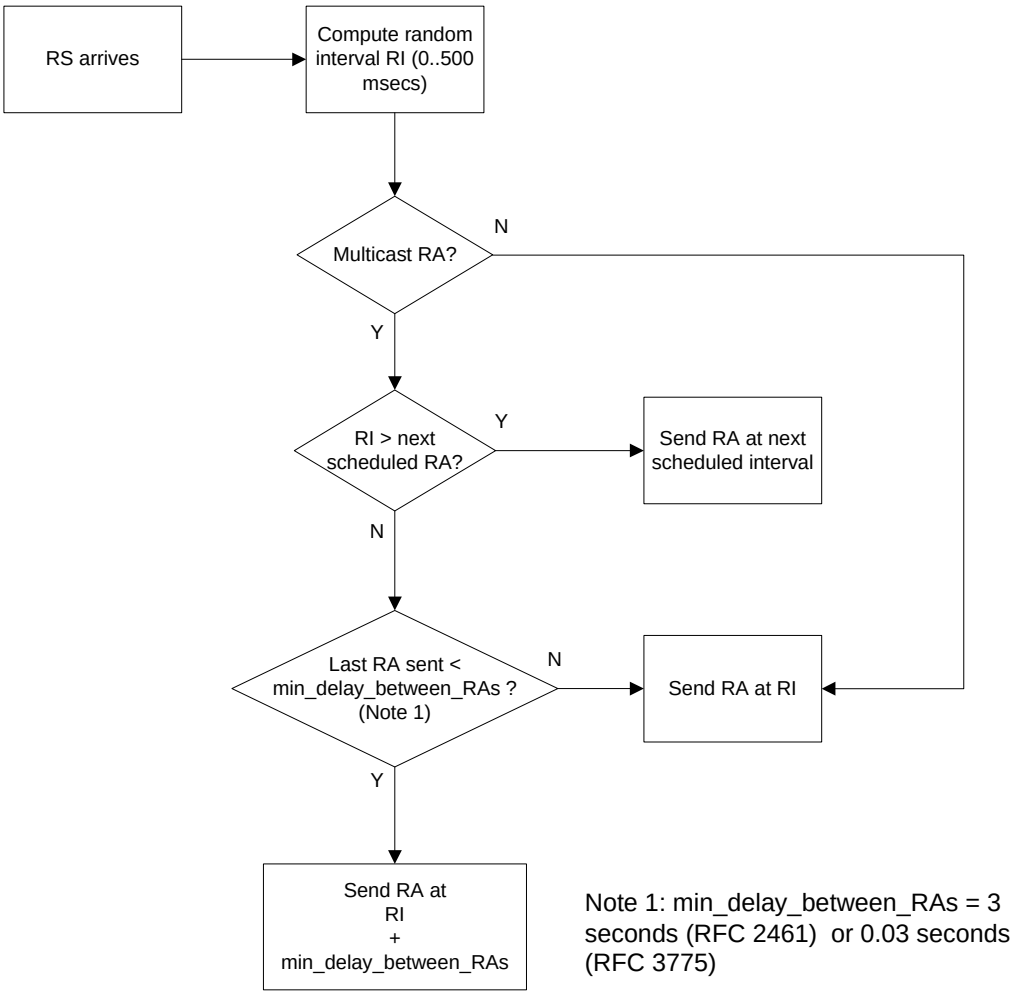

Figure 10 Processing Router Solicitations

The test experiment was as follows. The IPv6 capable VoD server was located at the Correspondent Node and streamed 1.5Mbps MPEG1 video clip of 30 seconds duration was streamed to the Mobile Node. At 10 seconds into the video clip the AP to which the MN was associated with was switched off, forcing a handover to the other network. At the end of the clip the handover latency and packet loss (reported by the VoD client) were noted. This was repeated 10 times for each value of RA interval configured on the Cisco routers. These RA intervals were 300ms (the RFC 3775 minimum), 1000ms and 3000ms.

	Avg Latency (seconds)	Avg # Packets Lost
300 ms	1.917	245.376
1000 ms	2.448	313.344
3000 ms	3.013	385.664

Table 1 Results of RA Interval Tests

It can be seen that changing the RA interval does not have as much effect on reducing the overall latency as we would like. This can be explained in that the rest of the handover procedure after receiving a RA, i.e. CoA configuration, DAD and CoA registration with the HA and CN is completely unaffected by reducing the RA interval. One can also see the number of packets lost in the video stream. On the client playback the stream would recover itself after handover but the break in the video and audio seemed about 1 or 2 seconds longer than the handover latency reported in the logs. It is easy to conclude that even tuning the RA interval to the lowest possible value will not suffice for real-time voice and video applications in a mobile environment.

4.7 Unicasting Solicited RAs

Another possible trick is to change the default behaviour of neighbour discovery so that a Router Solicitation is answered with a unicast RAs rather than the default multicast RA. In the standard algorithm depicted in Figure 10 it can be seen how a unicast RA only incurs the random delay interval and is not affected by the configured RA interval parameter (since this only applies to multicast RAs). To see what effect this would have on handover latency we had to replace a 2611 router with a linux PC based equivalent (since we were unable to configure the IOS accordingly).

	Avg Latency (seconds)	Avg # Packets Lost
Unicast RA	2.072	265.216

Table 2 Using Unicast RAs

From the table we can see that the results are slightly worse than the best we can get from configuring the RA interval. However, since the random interval is between 0 and 500 ms (the MAX_RA_DELAY_TIME constant in [3], we are unable to reduce this parameter further.

Want to conclude that even with tweaking the parameters we can tweak, MIPv6 handovers are simply not good enough to support real-time video and voice applications.

4.7.1 Eliminating DAD / Optimistic DAD

By removing the DAD procedure altogether we can reduce the handover latency even further (potentially by a second or so). However, removing this check altogether is not a realistic option both in terms of ratification by the IETF or by tuning an implementation's configuration. Thus, we are left with the option of fine tuning the DAD procedure in some way that reduces the time it takes for a MN to be able to use its CoA. A procedure called 'Optimistic DAD' which modifies [3] and [4] is proposed in [9], which essentially allows a CoA to be used before it has completed DAD. The CoA is marked as 'optimistic' as opposed to 'tentative' before completing DAD and is marked as 'preferred' once DAD is complete.

Unfortunately, we have not been able to source a suitable implementation of Optimistic DAD with which to test. An implementation will soon be made available from Monash University, but this will appear too late for the lifetime of the 6NET project.

5 Fast Handovers for Mobile IPv6

The aim of the Fast Handovers for Mobile IPv6 (FMIPv6) protocol [6] is to allow a MN to configure a new CoA, *before* it moves and connects to a new network. It also allows the MN to use the new CoA immediately upon connecting to the new network. Furthermore, the FMIPv6 protocol seeks to eliminate the latency involved during the MN's BU procedure by providing a bi-directional tunnel between the old and new networks while the BU procedures are being performed..

Thus, compared to normal MIPv6 operation, the FMIPv6 protocol claims to be more efficient in two respects:

1. It eliminates IPv6 configuration delay introduced by
    a. Router Discovery
    b. Address Configuration
    c. DAD
2. It removes the delay introduced by the MN performing BU procedures with its HA and CNs

5.1 Protocol Overview

At the core of the FMIPv6 protocol is the requirement for an access router (AR) to have knowledge of other ARs located close by that a MN may wish to connect to. Moreover, this also requires knowledge of the L2 Access Points that each AR is responsible for. Obviously, this requirement suggests that implementation of the protocol is most easily achieved for intra-organisational network deployments. Inter-organisational deployment will require some trust model being established between the relative organisations.

FMIPv6 introduces some new terminology:

- *AR*[1] – Access Router. The default router of the MN, i.e. the router to which the MR is currently connected to.
- *PAR* – Previous Access Router. The AR involved in handling a the MN's traffic prior to movement. The PAR is the router to which the Mobile Node is connected to before movement.
- *NAR* - New Access Router. The AR involved in handling the MN's traffic after movement has occurred. The NAR is the router to which the MN is connected to after movement.
- *PCoA* - Previous Care of Address. The Care of Address assigned to the MN before it moves.
- *NCoA* - New Care of Address. The Care of Address that is assigned to the MN after it has moved to a new network.

An overview of the new elements and a reference architecture is shown in Figure 11.

[1] Note that until the MN has moved AR = PAR and after movement AR = NAR

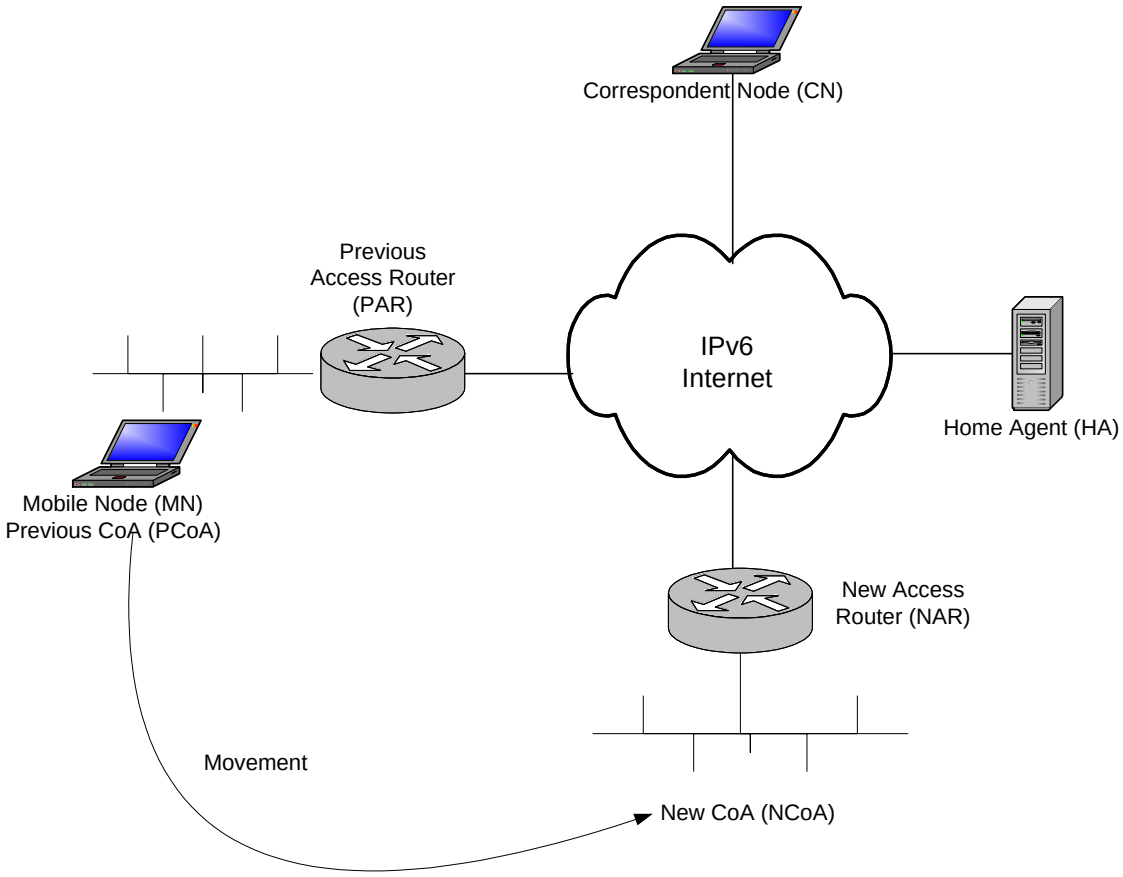

Figure 11 Overview of FMIPv6

The protocol also defines some new message types:

- RtSolPr - Router Solicitation for Proxy (from MN to PAR). Sent by the MN to request handover information from the PAR.
- PrRtAdv - Proxy Router Advertisement (from PAR to MN). Sent by the PAR to inform the MN of neighbouring links.
- FBU - Fast Binding Update (from MN to PAR). Sent by the MN to perform the Binding Update with a NCoA obtained from the PrRtAdv message.
- HI - Handover Initiate (from PAR to NAR). Sent by the PAR to the NAR to initiate the handover.
- HAck - Handover Acknowledgement (from NAR to PAR). Sent by the NAR to acknowledge the handover initiation.
- FBack - Fast Binding Acknowledgement (from PAR to MN). Sent by the PAR to acknowledge the FBU.

- FNA - Fast Neighbour Advertisement (from MN to NAR). Sent by the MN to announce its on-link presence to the NAR.

5.1.1 Mobile Node Initiated Handover

Figure 12 provides an overview of the handover procedure. For a MN initiated handover (i.e. it is the MN that takes the decision to move links) the MN issues a *RtSolPr* message to its current AR (PAR in the figure) in order to obtain information about its neighbouring networks. For 802.11 networks, the *RtSolPr* message will contain a list of APs that the MN can detect. The PAR replies with a *PrRtAdv* message which will contain a list of IPv6 layer information for each AR relative to each AP. This IPv6 information includes the link-layer addresses of the ARs and prefixes with which the MN can autoconfigure a CoA.

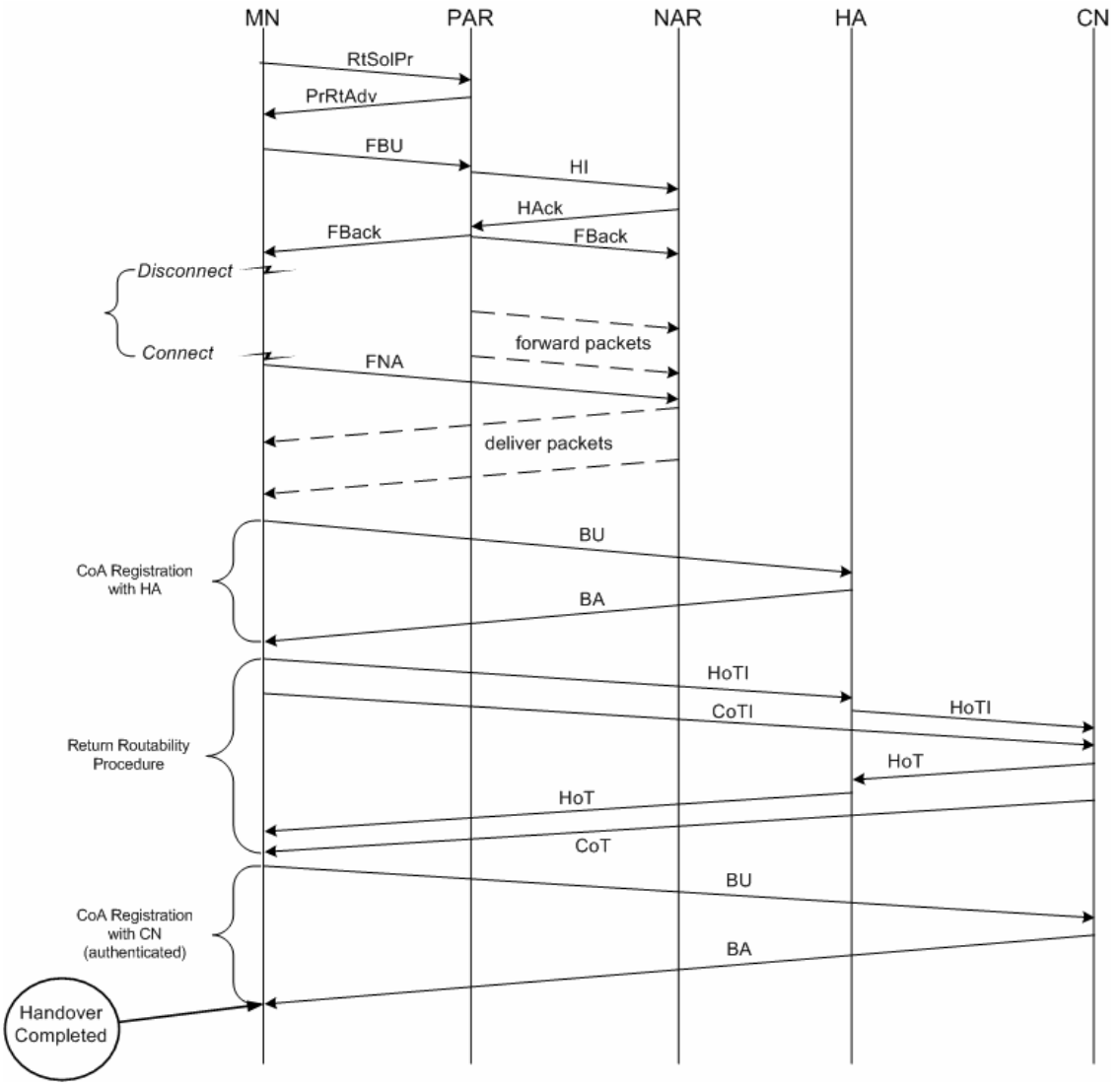

Figure 12 The FMIPv6 Handover Procedure

Upon receipt of the *PrRtAdv*, the MN can make a decision (e.g. based on 802.11 PHY signal strength information) as to which AP to associate with. The MN then sends a *FBU* to the PAR indicating which AP it is about to associate with (and thus which NAR it will connect to). The *HI* and *HAck* messages are to verify that the correct IPv6 configuration data is present. Upon receipt of the *HAck*, the PAR then establishes a binding between PCoA and NCoa and tunnels any packets bound for PCoA to NCoA

Thus, during movement, (i.e. once the *FBU* has been sent to the PAR by the MN), the PAR will forward packets for the MN's PCoA to the NCoA via the bi-directional tunnel. The NAR can buffer these packets until the MN arrives on its link and then deliver them to the MN. The MN announces its presence on the new link by sending a *FNA* message to the NAR. Once attached to its new link, the MN still uses the bi-directional tunnel and sends packets with source address = PCoA in the inner tunnel until it has completed the MIPv6 BU procedure. Note that the usual MIPv6 handover procedure for performing CoA registration with the HA and CNs occurs after the FMIPv6 procedure.

In this way, any packets normally lost during movement will be buffered by the NAR and delivered to the MN when it arrives on the new link. Furthermore, communication with CNs can continue via the bi-directional tunnel thus negating the usual latency effect for performing the MIPv6 BU procedure. Latency effects on real-time traffic will still exist, however they are reduced only to the time it takes to actually move (i.e. disconnect from the PAR and connect to the NAR).

5.1.2 Network Initiated Handover

In some network deployments, it may be possible for the network to initiate the handover procedure rather than the MN. One example scenario would be for an intelligent subsystem on the PAR to determine that a MN would be better served moving to another nearby network (e.g. due to it being topologically closer to its CNs or for traffic engineering purposes). In such situations, the PAR will send an unsolicited *PrRtAdv* to the MN containing the information with which the MN can connect to the new network. Apart from the absence of the initial *RtSolPr* message, the message exchanges are the same as in Figure 12. However, the processing is slightly different in that the MN must connect to the network indicated in the *PrRtAdv* by configuring a CoA for itself an issuing a *FBU* to the PAR.

5.1.3 Reactive handovers

The handovers discussed so far (both MN and network initiated) have assumed that the MN only moves to the new network once the FBU has been sent to the PAR. However, the situation can arise where the MN moves to the new network before it has had the chance to send the FBU to the PAR. In this case, the MN will send the FBU encapsulated inside the FNA that it sends to the NAR. The NAR will then forward the FBU to the PAR thus allowing the PAR to make the PCoA – NCoA binding and forward any packets destined for PCoA to NCoA. Of course, the time lag between the MN moving and the PAR receiving the FBU means that there is potential for packet loss during a reactive handover.

5.2 Analysis

Technically speaking, the handover latency using FMIPv6 will not be any quicker than for MIPv6. That is, if we define handover latency to be the time between losing connectivity from one network to resuming existing communications on the new network using the NCoA. This definition makes sense in MIPv6 as there is no way for the MN to resume communications until the NCoA has been registered with its HA. However, we have seen that FMIPv6 allows existing communications to continue throughout the entire handover process (assuming the NAR has sufficient resources to buffer packets until the MN attaches to the new link). Theoretically, the only effect on existing traffic flows for a predicted handover will be the latency involved when packets are buffered at the NAR. Thus, if we define this as the real, effective handover delay it can be expressed as:

$$t_h = t_{connect} - t_{disconnect}$$

with the latency experienced by packets being:

$$t_h = t_{FNA} + t_{deliver}$$

where t_{FNA} is the time it takes for FNA to complete and $t_{deliver}$ is the time it takes for the NAR to deliver the packets to the MN.

Thus for movement that can be predicted, a FMIPv6 handover *should not* result in any lost packets and any jitter for real-time streams will be minimised to the time it takes to perform the movement at L2 (a good rule of thumb being ~50ms for re-association in 802.11 networks, although OS and driver specific timers can push this time up considerably).

Of course this only holds true for predicted handovers. For reactive handovers there remains the possibility of packet loss since from the moment the MN disconnects from the old network, the PAR has nowhere to send packets destined for PCoA until it receives the FBU from the MN after it has arrived on the new network.

One of the open issues with the FMIPv6 is choosing when to tear down the bi-directional tunnel between the MN's PCoA and NCoA. Intuitively, this should be done once the MN has completed the MIPv6 BU procedure with all of it's CNs. However, the current FMIPv6 specification does not provide any signalling exchange for the MN to inform the PAR that it can stop forwarding packets. Thus, a soft state timer in the PAR set to a 'reasonable' value is the most likely solution that will be implemented.

Unfortunately, at the time of writing we do not know of any available implementation (compatible with a RFC 3775 compliant MIPv6 implementation) for FMIPv6 that we can test. It was hoped that the FMIPv6 implementation as part of the DAIDALOS project may have been made available for some collaborative tests. However, the implementation has not yet passed the integration testing of individual modules.

6 Conclusions

Attaining IPv6 addressing state in the current IP mobility management (Mobile IPv6) standard incurs by itself a significant transmission delay in the communications of the MN with its peers. This transmission delay is incurred with no involvement of AAA or QoS state establishment, for the MN at the new point of attachment.

Assuming real-world deployment of MIPv6 mobility management, for all contexts (i.e. IP addressing, Billing/Authentication, QoS provisioning), comprising the total IP connectivity state of the MN, the excessive transmission delay incurred, is due fundamentally to the reactivity in control signalling of the current IP mobility management model. For instance, in the case of IP addressing state the mobile node must perform a registration with its new point of attachment as soon as it has been detected on-link to the new point of attachment.

Under the current model of MIPv6, the observation may be safely generalised for both the establishment of billing/authentication and QoS provisioning state; that the visiting network, configures/establishes such IP connectivity state in reaction to the MN's detection on the new point of attachment; in real-world scenarios this perspective of IP mobility management incurs significant delay, large enough to impede any notion of transmission delay transparency to existing application connections. This is particularly important for novel interactive and real-time applications such as VoIP, multimedia streaming or network gaming.

We must therefore conclude that MIPv6, in its current form is not by itself sufficient to be the de-facto mobility management model in the mobile IPv6 Internet. Further optimisations relating to handover performance must be made in order to support interactive and real-time IPv6 applications in a mobile context.

Our tests have demonstrated that even with fine tuning the parameters of routers for optimum MIPv6 handover performance, we still do not approach anywhere near good enough handover times for real-time voice/video applications.

We have examined the fast handover protocol for MIPv6, FMIPv6. This aims to improve handover latency by eliminating IPv6 configuration latency and also prevents packet loss by the use of a bi-directional tunnel while physical movement and MIPv6 CoA registration are taking place.

To the best of our knowledge no implementation has yet been developed for us to perform handover tests. Yet this does not prevent us from reasoning that FMIPv6 will indeed reduce handover latency in almost all cases. In some cases, e.g. predicted handovers and relatively infrequent movement FMIPv6 promises to be sufficient for the real-time applications, most notably the killer mobile application VoIP. However, without being able to perform real tests it would be rather hasty to take this for granted.

References

[1] David B. Johnson, Charles Perkins, J. Arkko “Mobility Support in IPv6” IETF Internet Draft draft-ietf-mobileip-ipv6-24.txt, work in progress

[2] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, IETF RFC 2460, December 1998.

[3] T. Narten, E. Nordmark, W. Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, IETF RFC2461, December 1998.

[4] S. Thomson, T. Narten, “IPv6 Stateless Address Autoconfiguration”, IETF RFC 2562, December 1999.

[5] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”, IETF RFC 3315, July 2003.

[6] R. Koodli, “Fast Handovers for Mobile IPv6”, IETF Internet Draft draft-ietf-mipshop-fast-mipv6-03.txt, work in progress

[7] H. Soliman, C. Castelluccia, K. Malki, L. Bellier, “Hierarchical Mobile IPv6 mobility management”, IETF Internet Draft draft-ietf-mipshop-hmipv6-01.txt, work in progress.

[8] H. Jung, S. Koh, H. Soliman, J. Lee, K. El-Malki, B. Hartwell, “Fast Handover for Hierarchical Mobile IPv6 (F-HMIPv6)”, IETF Internet Draft draft-jung-mobileip-fastho-hmipv6-03.txt, work in progress

[9] N, Moore, “Optimistic Duplicate Address Detection”, IETF Internet Draft draft-ietf-ipv6-optimistic-dad-05.txt, work in progress.

[10] P. McCann “Mobile IPv6 Fast Handovers for 802.11 Networks”, IETF Internet Draft, draft-ietf-mipshop-80211fh-00.txt

[11] K. Suh, D. Kwon, Y. Suh, Y. Park, “Access Router Information Protocol (ARIP)”, IETF Internet Draft draft-suh-mipshop-arip-01.txt, work in progress.

[12] Y. Park, “Network-initiated Handover Framework for FMIPv6”, IETF Internet Draft draft-park-mipshop-netho-00.txt, work in progress.

[13] S. Park, E. Njedjou, N. Montavont, “L” Triggers Optimized Mobile IPv6 Vertical Handover: The 802.11/GPRS Example”, IETF Internet Draft draf-daniel-mip6-optimized-vertical-handover-00.txt, work in progress.

[14] S. Faccin, B. Patil, C. Perkins, F. Dupont, M. Laurent-Maknavicius, J. Bournelee, “Mobile IPv6 Authentication, Authorization, and Accounting Requirements”, IETF Internet Draft draft-le-aaa-mipv6-requirements-03.txt, work in progress.

[15] C. Vogt, R. Bless, M. Doll, T. K’fner, “Early Binding Updates for Mobile IPv6”, IETF Internet Draft draft-vogt-mip6-early-binding-updates-01.txt, work in progress.

[16] B. O’Hara, L. Yang, “Architecture for Control and Provisioning of Wireless Access points (CAPWAP)”, IETF Internet Draft draft-ietf-capwap-arch-00.txt, work in progress.

[17] N. Moore, J. Choi, B. Pentland, “Edge Handovers for Mobile IPv6”, IETF Internet Draft draft-moore-mobopts-edge-handovers-00.txt, work in progress.

[18] J. Arkko, B. Aboba, “Network Discovery and Selection Problem”, IETF Internet Draft draft-ietf-eap-netsel-problem-00.txt, work in progress.

[19] C. Williams, “Localized Mobility Management Goals”, IETF Internet Draft draft-ietf-mipshop-lmm-requirements-01.txt, work in progress.

[20] P. Tan, “Recommendations for Achieving Seamless IPv6 Handover in IEEE 802.11 Networks”, IETF Internet Draft draft-paultan-seamless-ipv6-handoff-802-00.txt, work in progress.

[21] J. Choi, “Fast Router Discovery with AP Notification”, IETF Internet Draft draft-jinchoi-l2trigger-fastrd-01.txt, work in progress.
[22] E. Njedjou, P.Bertin, P. Reynolds, “Motivation for Network Controlled Handoffs using IP mobility between heterogeneous Wireless Access Networks”, IETF Internet Draft draft-njedjou-inter-an-handoffs-00.txt, work in progress.
[23] A. Yegin, “Link Layer Triggers Protocol”, IETF Internet Draft draft-yegin-l2-triggers-00.txt, work in progress.
[24] G. Daley, J. Choi, “Movement Detection Optimization in Mobile IPv6”, IETF Internet Draft draft-daley-mobileip-movedetect-01.txt, work in progress
[25] K. El-Malki, H. Soliman, “Simultaneous Bindings for Mobile IPv6 Fast Handovers”, IETF Internet Draft draft-elmalki-mobileip-bicasting-v6-05.txt, work in progress.
[26] K. Baba, J. Cheng, R. Diaz, S. Mahidara, A. Mehta, V. Pandurangi, A. Singh, “Fast Handoff L2 Trigger API” IETF Internet Draft draft-singh-l2trigger-api-00.txt, work in progress.
[27] A. Yegin, E. Njedjou, S. Veerepalli, N. Montavont, T. Noel, “Link-layer Triggers and Hints for Detecting Network Attachments”, IETF Internet Draft draft-yegin-dna-l2-hints-01.txt, work in progress.
[28] S. Aust, N. Fikouras, C. Goerg, C. Pampu, “Policy Based Mobile IPv6 Handover Decision (POLIMAND)”, IETF Internet Draft draft-iponair-dna-polimand-01.txt, work in progress.

Glossary of Acronyms and Abbreviations

AAA	Authentication, Authorisation and Accounting
AAAH	Home AAA
AAAL	Local AAA
AP	Access Point
AR	Access Router
BA	Binding Acknowledgement
BSS	Basic Service Set
BU	Binding Update
CAP	Current Access Point
CoA	Care-of Address
CoT	Care-of Test
CoTI	Care-of Test Initiate
CN	Correspondent Node
DAD	Duplicate Address Detection
DHCPv6	Dynamic Host Configuration Protocol for IPv6
ESS	Extended Service Set
FBA	Fast Binding Acknowledgement
FBU	Fast Binding Update
FNA	Fast Neighbour Advertisement
HA	Home Agent
HACK	Handover Acknowledgement
HoA	Home Address
HoT	Home Test
HoTI	Home Test Initiate
HI	Handover Initiate
MN	Mobile Node
NA	Neighbour Advertisement
NAP	New Access Point
NAR	New Access Router
NCoA	New Care of Address

NS	Neighbour Solicitation
PAP	Previous Access Point
PAR	Previous Access Router
PCoA	Previous Care of Address
PrRtAdv	Proxy Router Advertisement
QoS	Quality of Service
RA	Router Advertisement
RR	Return Routability
RS	Router Solicitation
RtSolPr	Router Solicitation for Proxy
WLAN	Wireless Local Area Network