Seamless Handover in Heterogeneous Networks using SIP A Proactive Handover Scheme with the Handover Extension

Elin Sundby Boysen
Norwegian Defence Research Establishment (FFI)
UNIK - University Graduate Center Kjeller
Kjeller, Norway
Email: elin-sundby.boysen@ffi.no

Torleiv Maseng
Norwegian Defence Research Establishment (FFI)
Kjeller, Norway
Email: torleiv.maseng@ffi.no

Abstract

Mobile users move across different types of network, such as WiFi, WiMAX and UMTS. The mobile equipment is already capable of connecting to different network types simultaneously, but in such a heterogeneous environment, session continuity is still a challenge when changing connection from one network to another. The differences in properties on the physical layer and link layer promote higher-layer solutions for handover. Several architectures have already been proposed in either the network layer, in the application layer or as hybrid solutions. The application-layer-based Session Initiation Protocol (SIP) supports terminal mobility, but this procedure suffers from long handover delays. To ensure session continuity during handover we propose a new SIP extension, the Handover header that enables seamless handover. In the handover scheme we propose in this paper, we use the SIP Handover extension with a back-to-back user agent (B2BUA) that is deployed in the home domain of the mobile user. The handover can be assisted by either the B2BUA or the correspondent node.

Keywords: SIP; Mobility; Seamless handover; 3G; WiFi

1. Introduction

Modern users of laptops and other mobile equipment want to be online anywhere, anytime using the cheapest or most suitable network, and they expect the technical solution to be ready within a very short time. Many new phones and computers are already equipped with multiple interfaces that allow you to be connected to multiple networks at the same time, like handsets with UMTS and WiFi interfaces. However, a user that switches an ongoing session from one network to another, can experience delays and possibly even session loss. Such delays can be a nuisance when using applications like email or web browsers, but are particularly problematic for real-time sessions such as voice or video, where the user experience can be severely degraded due to the handover delays. How can seamless handover between different types of networks, so-called heterogeneous networks, be accomplished? Although the business side of this is still a matter of discussion, different technical solutions are definitely on their way. In this paper we look deeper into some of the challenges around mobility in heterogeneous networks and suggest an improved version of our Proactive Handover scheme using SIP [1] as one possible solution.

In heterogeneous networks we differ between vertical and horizontal handover. Horizontal handover is between two access points of the same kind, for example WiFi to WiFi handover. In this paper we will focus on vertical handover - the handover between access points of different types like for example UMTS and WiFi. When performing vertical handovers the mobility management protocol must not only provide location transparency, but also network transparency.

To obtain session continuity, there are two main approaches. Either to solve it on the network layer with Mobile IP or on the application layer with augmented existing protocols such as H.323 or Session Initiation Protocol (SIP) [2]. Both the network layer and the application layer approaches have their advantages and drawbacks, and the one does not necessarily exclude the other [3], [4]. We have chosen to focus on the application-layer approach because of its flexibility and ease of implementation.

The rest of this paper is structured as follows. First, in Section 2 we present issues concerning mobility in general and the basics of the SIP protocol. Then we present related work and describe the novelty of this contribution in Section 3. In Section 4, our Proactive Handover scheme is explained and we suggest some changes to it by introducing a new SIP Extension: the Handover header. The proposed solution is discussed and areas of future work are identified in Section 5. Finally, in Section 6 we make some concluding remarks.

2. Mobility and SIP

This section will provide the basic concepts of mobility in general, some of the challenges when considering Mobile IP for terminal mobility, and an introduction to Session Initiation Protocol (SIP).

2.1. Mobility

Mobility while providing session continuity has until recently been a virtue mainly reserved for operator-controlled mobile systems like GSM and UMTS. These systems consist of overlapping and interconnected base stations and support handover of calls between adjacent cells. A mobile node (MN) report a set of parameters, including their signal strength, and the base stations monitor the traffic load in their cells. The handover is managed by the network and initiated by the base stations.

The Internet had originally no support for session continuity during mobility. The IP addresses are used to describe the point-of-attachment (PoA) of a unit, ergo the location, not the identity of the unit itself. This is not a problem when the units attached to the Internet are stationary, but when the units are moving and need a new PoA, the IP address also usually needs to be updated. This makes session-oriented communication difficult. Mobile IPv4 [5] was introduced as a solution to this problem. Mobile IPv4 allows a node to move from one network to another while keeping the same home address. When the node is not in its home network it is given a care-of-address that is associated with the home address. The corresponding node (CN) uses only the home address. With IPv6 comes also Mobile IPv6 [6]. However, neither Mobile IPv4 nor Mobile IPv6 supports seamless handover. When changing PoA, the MN must realize that the connection with the first PoA is lost before it connects to the new. Lee et al. report average handover delays of 1896ms and 2470ms for two different test cases using Mobile IPv6 [7]. For real-time services this is too long. Maximum handover delays should ideally be less than 100ms, not more than 200ms [8].

Another challenge is that an operational infrastructure that supports Mobile IP must be in place. As these two important challenges –the handover delay and the need for infrastructure– are not yet properly solved; other solutions are being investigated concurrently. One of these is the Proactive Handover scheme. Instead of solving the handover problem on the network layer, proactive handover using SIP is proposed on the application layer. Proactive Handover was first presented in [1] and is explained in more detail in this paper.

A handover has three main phases: *handover initiation* when the mobile node searches for and discovers a new network, *handover preparation* when the node sets up a new link, and *handover execution* when the handover signalling take place and the connection is transferred to the new link. Using SIP we propose a solution to reduce and in most cases eliminate the delays due to each of these phases.

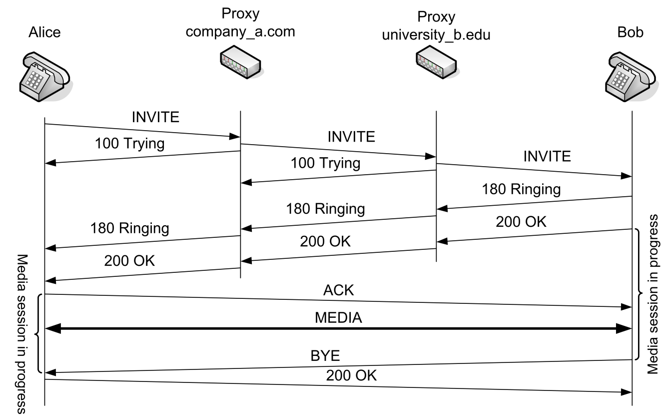

Figure 1. Standard setup of a session using SIP messages

2.2. Introduction to SIP

SIP is a protocol designed to establish, modify and terminate multimedia sessions. In short, SIP has been created to make it possible for end points to localize each other. It is an application-layer protocol and runs on top of transport layer protocols like TCP (Transmission Control Protocol), UDP (User Datagram Protocol) or SCTP (Stream Control Transmission Protocol). SIP works with both IPv4 and IPv6.

SIP is based on a request-response transaction model. A client sends a request and a server responds. A SIP message consists of the method name and set of header fields. The six basic methods are REGISTER, INVITE, ACK, OPTIONS, BYE and CANCEL. Later methods like REFER, NOTIFY, MESSAGE, SUBSCRIBE and INFO have been added. The SIP messages do not contain any information about the session itself. Instead, this information is provided in SDP (Session Description Protocol) messages contained in the SIP body.

Figure 1 depicts the signalling flow when setting up a session using SIP messages. In Figure 2 we can see an example of a message. This is a 200 OK response from Bob to Alice answering an INVITE request that was sent from Alice to Bob. In this message, the first line contains the response code, 200 OK. The rest of the message lines are header fields. The header field values of *To*, *From*, *Via*, *CSeq* and *Call-Id* are copied from the incoming INVITE request. In the *From* header field, Alice has included a *tag* parameter. Bob adds his tag parameter in the *To* header field. The two tags and the call id will together make the dialog identity. The three *Via* fields are added by Alice's SIP phone and the two proxies that forward the messages so that the response can take the same route back to Alice. In the *To* and *From* header fields the addresses are on a general form defining a logical recipient. Bob has added the *Contact* header field with his current location 'sip:bob@192.0.2.4' so that subsequent messages can be sent directly to him (as

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.university_b.edu
   ;branch=z9hG4bKnashds8
   ;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.company_a.com
   ;branch=z9hG4bK77ef4c2312983.1
   ;received=192.0.2.2
Via: SIP/2.0/UDP pc33.company_a.com
   ;branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@university_b.edu>
   ;tag=a6c85cf
From: Alice <sip:alice@company_a.com>
   ;tag=1928301774
Call-ID: a84b4c76e66710@pc33.company_a.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
```

Figure 2. Example of a SIP response (200 OK).

the ACK, BYE and 200 OK messages in Figure 1) and not through proxies.

A proxy on the initial route may require that it be in the signalling path throughout a session. In this case, the proxy will add the header field *Record-Route* and its IP address or a URI resolving to the address in the INVITE request. As the header field will also be copied into the 200 OK response, both end point will eventually know that messages should be routed through the proxy. The media packets, however, will still go directly between the end points.

SIP consists of different logical elements such as UACs (User Agent Clients), UASs (User Agent Servers), registrars, redirect servers, back-to-back user agents (B2BUA) and proxies. The end points are referred to as UAs (User Agents) and consist of a UAC and a UAS. Proxies can be either statefull or stateless. A stateless proxy server will only forward incoming requests and responses. A statefull proxy on the other hand, will maintain a state for each transaction, -that is which requests and responses belong to that transaction. Redirect servers receives requests and responds to the requester where it should send its request. The B2BUA is an element that can be described as a concatenation of a UAS and a UAC. It acts as a UAS when receiving a request and processing it. When forwarding the request to the corresponding node it acts as a UAC. Unlike a proxy server, it maintains not only transaction state, but also dialog state and *must* participate in all requests sent on the dialogs it has established. Often, B2BUAs also terminate and bridge the media streams to have full control over the whole session. This makes B2BUAs well suited for transcoding between two dialogs, to hide network internals, and for network interworking as it can have protocol adaptation.

SIP inherently supports personal mobility. This means that a user can be found using a single identifier regardless of which location or device (such as PCs, PDAs or phones) he or she is currently at [5]. Terminal mobility is more relevant when introducing wireless access and is the topic of this paper. Terminal mobility allows the user to move around with the device, and the device will roam between different IP subnets. We differ between pre-call mobility and mid-call mobility or in-session mobility. Pre-call mobility is the easiest part, as the MN will only need to re-register its new IP-address with the home registrar each time it changes IP-subnet. The focus on this paper is on mid-call mobility.

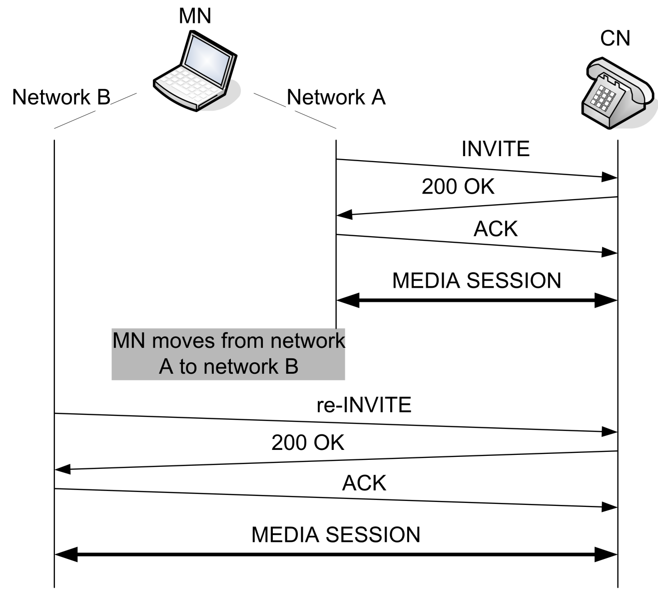

Figure 3. Setup and modification session according to [2] and [9].

The first suggestion for mid-call mobility support in SIP was presented by Wedlund and Schulzrinne in 1999 [9]. When the mobile node (MN) moves from one network to another it simply send a re-INVITE to its corresponding node (CN) telling it about its new IP address. This solution has been included in [2] and it is shown in Figure 3. When a new INVITE message is sent to change an existing dialog, a full description of the session is sent, not only a description of the changes. One challenge by using re-INVITE as described in [9] and [2] is that the handover delay when moving from one network to another can become too long. Referring to the three phases mentioned in Section 2.1, the MN will be disconnected through both the handover initiation (the MN must first realize that it has lost connection with the first network), the handover preparation (it must acquire a new IP address in the new network), and most of the handover execution phase where it sends the re-INVITE.

3. Related work and contributions

In this section we will first present some related work before we present the outline of the improved handover scheme presented in this paper.

3.1. Related work

Nakajima et al. [10] have analysed the handover delay for SIP mobility in IPv6. They measured handover delays from about 2s to 40s and have shown that this delay is mostly induced because of the Duplicate Address Detection (DAD) in IPv6. DAD imposes a delay from a user receives a Router Advertisement (RA) until the user can send its packets over the interface. The purpose of the DAD is to confirm the uniqueness of the autoconfigured IPv6 address on the link. With a modified kernel that omits DAD Nakajima at al still experience handover delays around 171ms for signalling and around 400ms for the media UDP packets. As in [10] Yeh et al. implements SIP terminal mobility over IPv6 [11]. Their solution also show very long handover delay due to DAD, reaching 1822ms before the media transmission is resumed. Without DAD they experience delays around 218ms for media resumption. The SIP mobility procedure in IPv4 network shows approximately the same delays as in IPv6 without DAD. In [12], Fathi, Chakraborty and Prasad studies SIP session setup delay over UMTS wireless networks with and without Radio Link Protocols (RLPs). They model and evaluate the protocol stacks SIP/UDP/RLP and SIP/TCP/RLP and show that the session setup delay (from INVITE is sent from UAC until ACK is received by CN) is lower for SIP over UDP than for SIP over TCP, around 4.6s using 9.6kbps bandwidth and 2.9s using 19.2kbps. The reason for considering these low bit rates, is that they assume the bandwidth allocated for SIP signalling in UMTS systems to be around this magnitude. These results confirm the results found by Wu et al. in [13], where handover between WLAN and WWAN is modelled. Here, a 128kbps channel in the UMTS network gives a handover delay of approximately 1.5s due to channel loss. For handover from WWAN to WLAN, the delay induced by for instance the DHCP address assignment is more important than the transmission delay of the new INVITE message over WLAN that is less than 1ms.

Several architectures and implementations have been suggested to overcome the challenges of too long handover delays. Some propose a combination of SIP and Mobile IP as Wang and Abu-Rgheff in [4], however most of the effort has been in providing new schemes and architecture to improve SIP.

We differ between *soft* ("make-before-break") and *hard* ("break-before-make") handover. In the hard handover all resources in the first connection are released before establishing a new connection. During soft handover, the equipment is able to communicate over multiple interfaces and thus using resources in both networks simultaneously. Some examples of hard handover are [2], [14], [15] and [16] and an example of soft handover is presented in [17]. Chahbour et al. [14] put forward Hierarchical Mobile SIP enforced by a predictive address reservation to reduce handover delay. Banerjee et al. suggest in [17] to let each base station in each of the wireless technologies (GPRS, CDMA, WLAN etc) be equipped with a back-to-back user agent (B2BUA). On the initiation of a handover the B2BUA of the old access network duplicates the incoming RTP packets and sends them to the B2BUA in the new access network. When the mobile node receives packets through the new interface, it releases the old B2BUA. Another solution that also suggests new entities in the subnets is presented in [15]. Here, Bellavista et al. introduce application-layer middleware to support session continuity. Their Mobile agent-based Ubiquitous multimedia Middleware (MUM) described in [15] and [18] consists of a Proxy Switch (PS) at the ingress of each domain and a Proxy Buffer in each subnet. A Handover Agent Activator (HAA) present in each subnet can activate a Handover Agent (HA) in conjunction with a B2BUA in the Proxy Buffer when a MN enters the subnet. The solution supports both vertical and horizontal handover and is very relevant for data streaming. Packets are being buffered in both the old and the new domain ensuring that no packets are lost while the MN is disconnected during the actual handover. While this solution ensures zero packet-loss, the disconnection time may be problematic for real-time sessions. Tsiakkouris and Wassell suggest in [16] to use location information from the Access Point Location Protocol (APLP) in combination with SIP to anticipate handovers and thus reduce handover delays. By introducing a SIP Mobility Anchor Point in the different domains that can forward media packets from the old to the new address while the MN informs the CN about its new location, packet loss can be avoided.

IEEE 802.21 [19], Media Independent Handover (MIH) is an emerging standard created originally to support handover and interoperability in heterogeneous networks consisting of different technologies in the IEEE 802-series. Later, handover between 802 technologies and non-802 technologies like cellular systems has also become part of the standardization work in 802.21. The scope of IEEE 802.21 is to assist in handover performed on layer 3 and above. It provides link layer triggers or events describing changes in link state or link quality, and network information about available networks and neighbour maps. It can also provide information about load balancing. A node can use the information to perform its own handover procedure or it can in some cases use the 802.21 specific handover commands. 802.21 also makes network initiated handover possible. The use of the 802.21 framework requires that the access points can provide link information through MIH messages. Containers for these messages are currently defined in 802.11u and 802.16g.

3.2. Improving SIP Handover

Many of the mentioned solutions for SIP handover provide a faster handover that reduces the handover delay or packet loss. Some of the solutions also introduce new network elements that must be present in either access points or in the subnets. We wish to provide a handover solution that is easily implemented and deployed, and that can support both soft and hard handover. We do not assume that we can deploy network elements in all the subdomains our user might enter, and rely instead on a B2BUA in the home network. We have already introduced our Proactive Handover scheme in [1]. It is summarized here and we propose a new header field, the *Handover* header, to improve the handover scheme. As the SIP protocol is flexible and easy to extend, we suggest this extension that is more in line with the SIP notation than what was previously suggested in [1].

4. Proactive handover

In this section we will first give a short summary of the proactive handover scheme as we described in [1]. Then we discuss different ways of triggering the handover before we suggest some improvements to the existing scheme.

4.1. The existing scheme

The purpose of the proactive handover scheme is to provide the means to support vertical handover that does not require changes in existing SIP infrastructure and is easily deployable. We assume that the end node user equipment has more than one interface carrying IP traffic, for instance WiFi (IEEE 802.11), Ethernet (IEEE 802.3) and UMTS, and that the user can obtain network access through these simultaneously. We also assume that the user can reach its CN through routes via each of these, i.e. more than one route. The scheme is backwards compatible, meaning that if a MN that is prepared for proactive handover finds that it is communicating with a node that does not support proactive handover, the MN will fall back to the ordinary SIP behaviour.

To promote the success of terminal mobility with SIP, four points are of importance:

1) *Delay:* The handover delay must be short enough not to break an ongoing session or to introduce serious degradation of user experience during the handover. This is especially important in real-time sessions.
2) *Packet loss and jitter:* The packet loss and jitter during the handover should be minimized. In addition to a degraded user experience, too high packet loss or jitter can make it impossible for a streaming session to synchronize and thus interrupt the whole session.
3) *Recovery capabilities:* A good handover scheme should in the case of sudden link loss recover fast enough to prevent sessions from collapsing.
4) *Ease of deployment:* To ease deployment of a handover scheme, the possibility of gradually deployment should be supported.

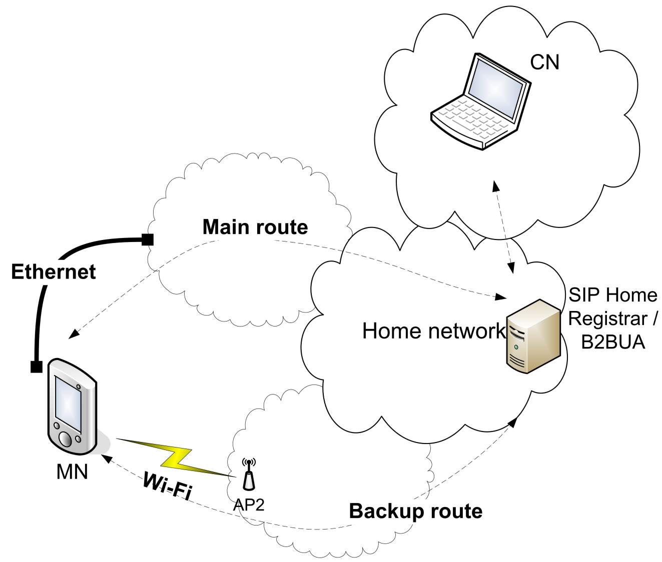

Figure 4. Reaching the CN through two possible routes

As in several of the handover architectures presented earlier, also our scheme suggested in [1] requires the use of a B2BUA. The B2BUA is situated in the mobile node's home network and bridges calls between the MN and its CN. As the B2BUA not only intercepts the signalling messages but also the media transport, the CN does not need to support proactive handover and the two call legs may even use two different codecs. The MN can still switch between different networks as the handover is managed between the MN and the B2BUA. This is depicted in Figure 4.

In the home network there is also a registrar. When the MN registers with the registrar, it registers its main address but also the current address of the backup interface(s). The interfaces on which the MN can be reached are prioritized by adding a parameter *if_q* in the *Contact* header field of the REGISTER request. To make sure that the registering of the backup interface does not overwrite the first register, the parameter *ua_id* is also added to the *Contact* header field. *ua_id* is a random string provided by the user agent and is the same for all main or backup registrations until the user agent re-registers or unregisters. Registering a backup interface can also be done once a session has already been established. When receiving a REGISTER message with the *ua_id* and *if_q* parameters, the registrar will provide a *if_no* parameter in its 200 OK response to the MN. The *if_no* parameter tells the MN how many interfaces the registrar has currently registered. If the registrar does not support proactive handover, it will only ignore the *ua_id* and *if_q* parameters and will consequently not reply with a *if_no* parameter in its Contact header field. This tells the MN that the registrar does not support proactive handover and it will continue its communication as any other SIP node. The message exchange for the register process is shown in

Figure 5.

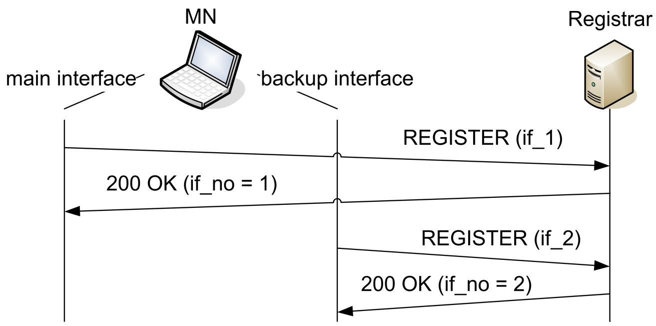

Figure 5. Register two interfaces.

Once a session has been initiated, the MN uses its backup interface to send a new INVITE message addressed to the CN. This message uses the same call-id as the ongoing session, but one of its SDP attributes is set to *sendonly*. The B2BUA will intercept the message and recognize the INVITE message as a backup INVITE. The B2BUA does not initiate a new call leg towards the CN, but replies with a 200OK to the MN. Thus a backup session has been initiated. No media packets will go through this route until a handover is triggered.

When the handover is triggered in the MN it will send a new INVITE request through the backup interface with SDP attribute set to *sendrecv* and the B2BUA will route all packets through this interface. The B2BUA can start sending media over the new interface as soon as it receives the re-INVITE and does not have to wait for the full INVITE - 200 OK - ACK three-way handshake to complete. Figure 6 shows the message exchange for the initialization of the session and the handover. The session is initiated using the main interface. The backup session is initiated, but no media packets go through the backup interface. When the handover is triggered, a new INVITE message is sent over the backup interface and the session is activated. The B2BUA bridges the call legs between the CN and the backup session. When the handover has taken place the register is updated so that the interface that was the backup interface is now set up as main interface.

4.2. Triggering the handover

As we have previously defined handover in three steps along a timeline (handover initiation, handover preparation and handover execution), it is important also to mention the handover decision algorithm that triggers the whole operation, and the handover metrics on which the handover decision algorithm bases its outcome.

Proactive handover using SIP [1] only describes how to prepare for and perform the handover. The handover must be triggered by some event that is not specified in the proactive

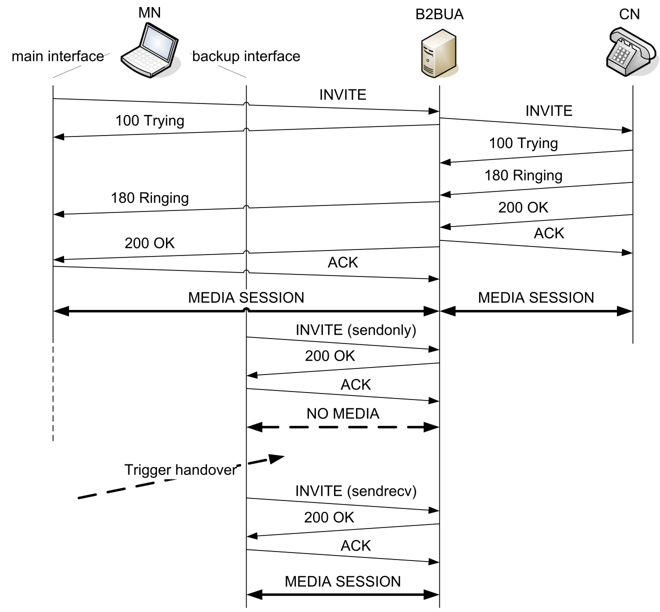

Figure 6. Initiating the session and the backup session.

handover scheme. SIP is an application layer protocol and is independent of the type of network that carries the messages. It is important to maintain this independence also when suggesting extensions to SIP. However we wish to use SIP for mobility management by using the handover procedure above. By keeping the handover decision algorithm as a separate procedure, we ensure that Proactive Handover can be used over different network types.

There can be many different reasons why we want to perform a handover. The user can have entered the coverage of a cheaper or otherwise better network to which he also has access. The most obvious reason is of course that the underlying transport link is broken, overloaded or deteriorating. In a heterogeneous network handover metrics should include relative signal strength, link quality, user preference, network conditions, application types and cost.

The link quality can be monitored by measuring frame or bit-error rate, packet jitter, delay and packet loss, as well as signal strength in wireless networks. However, these parameters will have different characteristics depending on the underlying network. A decision to trigger a handover must be made on the basis of which type of network is currently utilized. The cause of a handover in one network type may not be the best for another.

As the proactive handover scheme does not define the handover decision algorithm, this can very well be based on the 802.21 framework mentioned in Section 3.1. However, while we wait for the 802.21 framework to be deployed, any other way of triggering the handover can be used. Other possible solutions include the handover decision algorithm based on location data through GPS and APLP presented in [16]. If the communication takes place in autonomous networks like for instance military networks for tactical communications, one can also envision proprietary solutions based on a mix of public standards and tactical protocols. For instance is the ICMP (Internet Control Message Protocol) Source Quench protocol commonly used in military radios to request the sender to decrease the traffic rate of messages. In practice this means that the operating link has deteriorated. Thus this information can be used as a metric for the handover decision algorithm.

4.3. Improved handover scheme

Wu et al. [13] and Fathi et al. [12] point out the problems of long session setup delays, especially when connecting via links like UMTS radio links. Wu et al. show that the data connection setup delay can be in the range of 1500ms. This delay occurs even before the SIP signalling begins, during the data link setup. On the other hand Fathi et al. show that the SIP signalling itself can be very slow. In a normal session setup, the CN can begin sending media packets as soon as it has processed the INVITE message and the corresponding SDP. The MN can start its media session right after it has received and processed the 200 OK from the CN. However, if the conditions are as suggested in [12], that the signalling can have higher delays than the media packets, we risk that MN loses the first packets sent from the CN as it has not yet received the 200 OK from the CN or is still busy processing the 200 OK. If we again consider that the handover delay ideally should be in the range of 50ms - 200 ms, we argue that soft handover techniques must be used so that packets still can flow through the old interface while we wait for the new media route to be set up end-to-end. On the basis of this we propose how the Proactive handover scheme should support soft handover.

As mentioned in [1], one drawback of using a B2BUA to bridge the media streams between the MN and the CN is that it may become a vulnerable hot spot and also become a challenge in terms of scalability. One of the strengths of SIP is indeed that the signalling and the media can take different paths. The main argument for using a B2BUA to bridge both signalling and the media is that the CN does not need to support proactive handover. However, if the CN *can* support proactive handover, it would be better to let the media go directly between the two, making the CN duplicate the packets. While leaving more of the handover duties on the end points, we still want to provide the opportunity for media handling and handover management in the B2BUA for sessions where the CN does not support Proactive handover. To manage this, the B2BUA must know when to bridge the call and when not to.

In the following we suggest some changes to the scheme presented in [1]. We still want to provide a solution that is easy to deploy and that can be managed with only a few changes to the UA requiring the handover and to the B2BUA that assists in it. At the same time we want to improve the previous scheme by utilizing information already available through existing RFCs. We want seamless handover through the use of multiple interfaces that are active concurrently, and a means to make the UAS either in the CN or in the B2BUA start duplicating the media packet. This can be achieved by using a SIP extension, the *Handover* header that we propose here.

The *Handover* header bears resemblance to the *Join* header field defined in [20] and the *Replaces* header defined in [21]. When MN has obtained a new data connection through the backup interface, it sends an INVITE message over the backup interface. This INVITE initiates a new dialog with the CN. The CN replies with the usual 200 OK / ACK hand-shakes. A Handover header field could look like this:

```
Handover: a84b4c76e66710@pc33.company_a.com
;to-tag=a6c85cf
;from-tag=1928301774
```

In this Header field the MN has included the original *Call-Id*, *to-tag* and *from-tag* so that the CN can identify the right dialog.

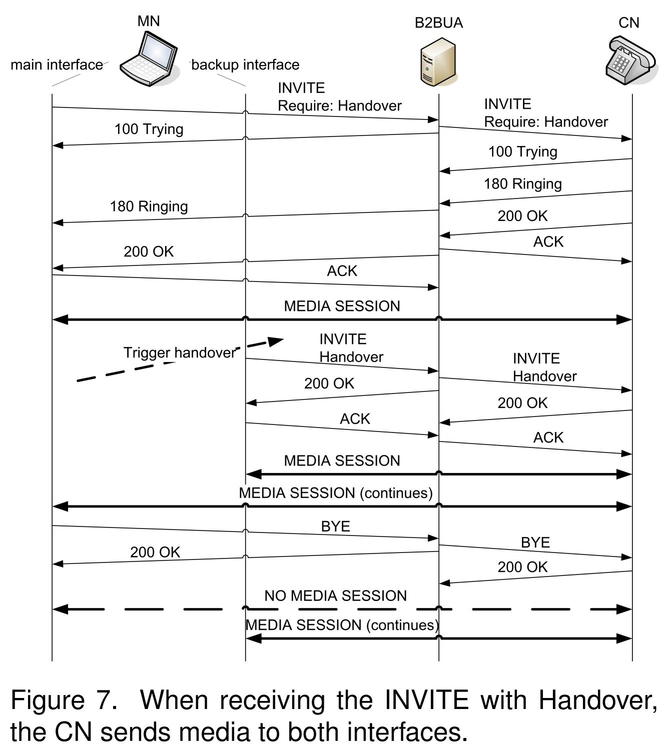

Figure 7. When receiving the INVITE with Handover, the CN sends media to both interfaces.

As described in 7, the CN then start to send the media packets to both interfaces and responds with a 200 OK. As the MN will now receive the same media packets from two directions, duplicate packet detection and filtering is necessary in the mobile node. When the MN sees that the new media stream is of a certain quality, it sends a BYE message over the initial interface and thus stops handling the incoming media packets. The CN will stop duplicating media packets as soon as it has processed the BYE message and then it sends the 200 OK response.

The B2BUA must know when to bridge a media stream and when to leave the handover management to the end points as described in Figure 7. In the initial INVITE the MN uses the *Require* header field listing the option tag *handover*. The B2BUA will forward the INVITE with the *Require* to the UAS in the CN. In the cases where a UAS does not support the extension listed in the *Require* field, the RFC3261 [2] states that the UAS *must* respond with status code 420 Bad Extension and add the *Unsupported* header field where it lists the unsupported extensions required by the UAC. Upon receiving a status code 420 Bad Extension, the B2BUA knows that it will be responsible for bridging the media. This can also occur if the CN realises that it does not have enough resources to handle an eventual handover. The B2BUA sends a new INVITE to the MN without the *Handover* header field. This time the Contact field is changed to indicate the B2BUA's address. When the CN responds with its 200 OK, the media path is set up between the CN and the B2BUA and between the B2BUA and the MN. This is shown in Figure 8. If the CN however

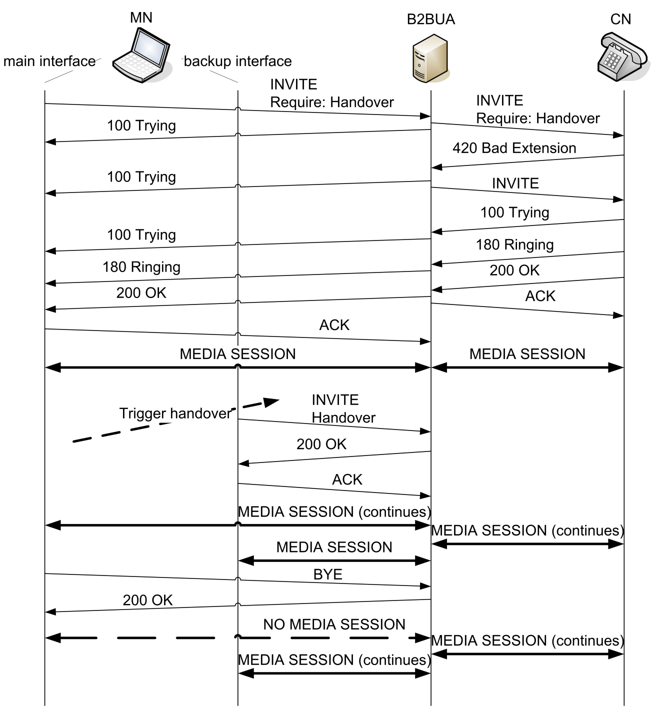

Figure 8. When receiving the INVITE with Handover, the B2BUA sends media to both interfaces. The CN knows nothing about the handover.

does support the *Handover* extension, it will answer with a 200 OK and the media path can be set up directly between the MN and the CN as in Figure 7.

The *Handover* extension must require that the sender authenticates himself. The mentioned RFCs [20] and [21] propose mechanisms for this, as all three methods can be a security threat in the form of call hi-jacking. Digest authentication or end-to-end message integrity such as S/MIME are used. The need for authentication was one reason to propose a passive backup session in [1]. The B2BUA could require the backup interface to be authenticated before setting up the backup session. However, as the UA on the mobile node already knows the credentials used in the main session, these can be reused when setting up the backup session.

There may be situations where the original link is so unstable that the MN does not have enough time to initiate the new dialog, perform packet duplication detection, filtering and synchronisation, in addition to closing the dialog using BYE over the old interface. The UAC will then send the BYE message using the new interface to release the resources spent on packet duplication in the UAS that handles the handover management.

Should the main link be broken before we have the time to initiate the handover, a regular INVITE will be sent over the backup interface using the existing dialog id and the *handover* extension will not be used.

4.4. Changes from the old to the new Proactive Handover scheme

In comparison with the scheme presented in [1] some changes are more salient than others. In [1] we proposed to register the backup interfaces with the registrar in the B2BUA either before we initiated a session, were they known, or during the session as they became available. Here we suggest that the register process is kept apart form the B2BUA logic as they are indeed defined as different logical entities. The handover shall be completed successfully before the registrar is updated. We do this because a new register message only will be relevant for subsequent sessions. The UAS that handles the handover in the ongoing session will only use the address it has found in the *Contact* header field and is informed of the address change through the new INVITE message. By only contacting the registrar when an address change actually has occurred, this also saves some unnecessary REGISTER transactions.

In this new scheme, one has the opportunity accept the incoming request and to set the media direct on hold with one of the SDP attributes set to *sendonly* as it is done in [1]. However, here, this SDP attribute is not used to determine whether this is a backup session or not, as we use the *Handover* header instead. The reason for suggesting the directly-on-hold solution in the previous solution was that in case of a sudden break of the main link, the re-INVITE would theoretically be quicker than setting up the session from the beginning. If we would have to authenticate the

MN when making the backup session, this would be true. In this new proposal we provide credentials for authentication in the re-INVITE and do not necessarily need the challenge/response mechanism that requires several messages back and forth between the MN and the CN or B2BUA. This means that (given that the authentication is accepted) an INVITE message activating a passive backup session will not be processed faster than the first INVITE request that puts the backup session on hold. Thus, unless a passive session is used to monitor and compare which connection provides the best link properties, a re-INVITE initiating a backup session should only be sent when the actual handover will take place.

5. Discussion and future work

With the proposed Handover scheme, we provide a solution for seamless handover in heterogeneous networks. Given that a new session is set up before the old is released, handover delays can be avoided because there will always be at least one functional path between the MN and the CN. Sources of disturbance in the media flow can can occur in the process of handling duplicated RTP packets arriving over two different interfaces and smoothing out any differences in path delays. However, the delays due to differences in the two path delays (the time between CN and MN) is expected to be less than any handover delays occuring due to a break-before-make scheme. In continuation of the proof-of-concept implemented for the proactive handover scheme presented in [1] the new solution is under implementation.

We assume that we can rely on the lower layer mechanisms, instructed by the application, to create a new data link connection based on a set of rules (Examples: ”Set up WLAN connection if I am currently on a UMTS connection as the WLANs are usually faster and cheaper” or ”Whenever I am connected using Ethernet and I discover an accessible WLAN AP, I shall set up a data link connection as long as I am not low on battery”). As we can prepare for the handover by setting up the new data connection while the old is still active, a decision on how much resources are to be used during the handover has to be made by the handover decision algorithm. If we, for instance, wait with the search for and setup of new data links until we actually have an ongoing session, the MN can save unnecessary searches and updates of the backup interface. This saves battery in the MN, but also resources in the network used for backup. This solution is not applicable, though, as it requires a common set of rules for all the applications on the MN that require network access, not just the SIP UA.

We have described a solution that support the use of the CN as the handover assistant when duplication media packets. When the B2BUA tries to find out whether the CN supports the Handover extensions, it will get a 320 Bad Extension in response if the CN does not support Handover. This results in a longer setup delay at the initiation of the session. Further studies will show to what degree this extra delay in the beginning degrades the user experience.

As already mentioned, security will be a very important issue when implementing the Handover extension. This will also be subject to further study, as it is necessary to study whether the security mechanisms suggested in Section 4.3 are good enough to prevent call hi-jacking.

6. Conclusion

Mobile users move across different types of network, such as WiFi, WiMAX and UMTS. The mobile equipment is already capable of connecting to different network types simultaneously, but in such a heterogeneous environment, session continuity when changing connection from one network to another is still a challenge. The differences in properties on the physical and link layer promote higher-layer solutions for handover. In this paper we have presented various challenges when handling handover in heterogeneous networks and some of the solutions proposed to overcome them. The application-layer-based Session Initiation Protocol (SIP) supports terminal mobility, but this procedure suffers from long handover delays. Various architectures and procedures have been proposed to manage handover in SIP. However, solutions proposed so far mainly considers reducing the handover delay when disconnecting from one access point and connecting to the new. Some also suggest deploying network elements such as B2BUAs in all the subnets to assist during a handover.

We propose a new SIP extension, the *Handover* header field, which enables seamless handover. The MN will connect to a new access point and set up a data link while the first interface is still connected and a session is active. During the handover period, the MN holds two concurrent sessions to the same B2BUA and receives media packets on both the old and the new data link before the old session is released. If the CN also support the *Handover* extension, the media path can go round the B2BUA and thus reduce the load on the B2BUA. The solution can easily be implemented using a B2BUA in the mobile node’s home domain. Thus, as an example, a VoIP provider can offer his mobile customer support for handover, independent of which domain the he is currently visiting.

References

[1] E. S. Boysen, H. E. Kjuus, and T. Maseng, “Proactive handover in heterogeneous networks using SIP,” in *Proceedings of the Seventh International Conference on Networking 2008 (ICN 2008)*. IEEE, April 2008, pp. 719–724.

[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, “SIP: Session Initiation Protocol,” RFC 3261 (Proposed Standard),

Jun. 2002, updated by RFCs 3265, 3853, 4320, 4916. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[3] H. Lee, S. W. Lee, and D.-H. Cho, "Mobility management based on the integration of mobile IP and session initiation protocol in next generation mobile data networks," in *IEEE 58th Vehicular Technology Conference (VTC), 2003.*, vol. 3. IEEE, October 2003, pp. 2058–2062.

[4] Q. Wang and M. A. Abu-Rgheff, "Mobility management architectures based on joint mobile IP and SIP protocols," *IEEE Wireless Communications*, vol. 13, no. 6, pp. 68–76, December 2006.

[5] C. Perkins, "IP Mobility Support for IPv4," RFC 3344 (Proposed Standard), Aug. 2002, updated by RFC 4721. [Online]. Available: http://www.ietf.org/rfc/rfc3344.txt

[6] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775 (Proposed Standard), Jun. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3775.txt

[7] J. S. Lee, S. J. Koh, and S. H. Kim, "Analysis of Handoff Delay for Mobile IPv6," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall*, vol. 4. Los Angeles: IEEE, September 2004, pp. 2967–2969.

[8] ETSI, "TS 101 329 -2 v2.1.3 (2002-01) Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON systems; Part 2: Definition of speech Quality of Service (QoS) classes," www.etsi.org, 2002. [Online]. Available: www.etsi.org

[9] E. Wedlund and H. Schulzrinne, "Mobility Support using SIP," in *WOWMOM '99: Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*. New York, NY, USA: ACM Press, 1999, pp. 76–82.

[10] N. Nakajima, A. Dutta, S. Das, and H. Schulzrinne, "Handoff delay analysis and measurement for SIP based mobility in IPv6," in *IEEE International Conference on Communications, 2003. ICC '03.*, vol. 2. Convent Station, NJ, USA: IEEE, May 2003, pp. 1085–1089.

[11] C.-H. Yeh, Q. Wu, and Y.-B. Lin, "SIP Terminal Mobility for both IPv4 and IPv6," in *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS)*. IEEE, July 2006, pp. 53–53.

[12] H. Fathi, S. S. Chakraborty, and R. Prasad, "Optimization of SIP Session Setup Delay for VoIP in 3G Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1121–1132, September 2006.

[13] W. Wu, N. Banerjee, K. Basu, and S. K. Das, "SIP-based vertical handoff between WWANs and WLANs," *IEEE Wireless Communications*, vol. 12, no. 3, pp. 66–72, June 2005.

[14] F. Chahbour, N. Nouali, and K. Zeraoulia, "Fast Handoff for Hierarchical Mobile SIP Networks," *International Journal of Applied Science, Engineering and Technology*, vol. 5, pp. 34–37, 2005.

[15] P. Bellavista, A. Corradi, and L. Foschini, "SIP-Based Proactive Handoff Management for Session Continuity in the Wireless Internet," in *26th IEEE International Conference on Distributed Computing Systems Workshops 2006, (ICDCSW06)*. IEEE Computer Society, July 2006, pp. 69–69.

[16] S. Tsiakkouris and I. Wassell, "PROFITIS: Architecture for Location-based Vertical Handovers Supporting Real-Time Applications," in *25th IEEE International Performance, Computing, and Communications Conference, 2006 (IPCCC 2006)*. IEEE, April 2006, pp. 629–634.

[17] N. Banerjee, S. K. Das, and A. Acharya, "SIP-based Mobility Architecture for Next Generation Wireless Networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. IEEE Computer Society, March 2005, pp. 181–190.

[18] P. Bellavista, A. Corradi, and L. Foschini, "Application-Level Middleware to Proactively Manage Handoff in Wireless Internet Multimedia," in *Management of Multimedia Networks and Services (MMNS)*, vol. 3754. Berlin / Heidelberg: Springer, October 2005, pp. 156–167.

[19] IEEE 802.21, "http://www.ieee802.org/21/index.html," Web page, 2003. [Online]. Available: http://www.ieee802.org/21/index.html

[20] R. Mahy and D. Petrie, "The Session Initiation Protocol (SIP) "Join" Header," RFC 3911 (Proposed Standard), Oct. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3911.txt

[21] R. Mahy, B. Biggs, and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header," RFC 3891 (Proposed Standard), Sep. 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3891.txt