Optimizing SIP Application Layer Mobility over IPv6 Using Layer 2 Triggers

Emil Ivov
Network Research Team
Louis Pasteur University / LSIIT
Illkirch, Strasbourg
emcho@clarinet.u-strasbg.fr

Thomas Noël
Network Research Team
Louis Pasteur University / LSIIT
Illkirch, Strasbourg
noel@clarinet.u-strasbg.fr

Abstract— **This paper describes a solution for the optimization of Session Initiation Protocol (SIP) based mobility over IPv6 in an 802.11b network as well as its theoretical evaluation and actual performance. Time necessary for a complete handoff depends on multiple factors. In this paper we focus on the delay accumulated during movement detection and try to bring it to a minimum using upward propagation of events generated by a Layer 2 entity. This allows us to speed up handoffs and get closer to a mobility solution applicable in "real-world" communications. We are using SIP both as a communications signalling protocol and as a means of handling application layer terminal mobility.**

Keywords-component; mobility; SIP; IPv6; WLAN; 802.11b; handover; triggers

I. INTRODUCTION

The proliferation of wireless devices along with the rapid growth of the Internet is demanding the Internet community to move from Internet Protocol version 4 (IPv4) [1] to Internet Protocol version 6 (IPv6) [2]. The major motivation behind this is the limitation of the IPv4 address space. Although Network Address Translation (NAT) is widely used to circumvent the address space problem, it fails to provide the global routability. IPv6, on the other hand, is designed to solve such problems. The expanded address space offered by IPv6 will enable assignment of globally routable IP addresses to every possible device connected to the Internet.

Wireless network access is increasingly popular. Wireless communications offer numerous advantages such as movement during a session, and network access at a fair rate among nodes. Mobility between access points that are part of the same subnet is managed by layer 2 mechanisms. When a Mobile Node (MN) connects to an Access Point (AP) in another subnet, however, the IPv6 address of the MN becomes topologically invalid. This kind of movement has to be managed by upper layer protocols.

Mobile IPv6 [3] is one such protocol. It is designed to maintain network connectivity for hosts roaming across the Internet. It allows mobile nodes to manage global communication through a home address and yet maintain packet flow while away from home using a Care-of-Address for each different point of network access. Another way to provide terminal mobility is using an application layer protocol such as SIP for example. This is achieved by re-initializing all sessions, active at the time an MN moves between subnets. The correspondent application gets notified for a mobility event and received an MN's new IP address through a session re-initialization request.

Many recent studies such as [13], [14], and [15] describe different handover optimizations when using a layer 3 protocol such as Mobile IPv6. A commonly employed technique is using information (messages) available in lower layers. The method is often referred to as Layer 2 or Cross Layer Trigger usage [10]. In this paper we propose and evaluate a way to use these triggers in an application layer mobility solution based on SIP.

The rest of the paper is organized as follows: Sections II.A and II.B describe SIP and SIP based terminal mobility. Section II.C presents Wireless LAN [6] and general mobility. Section III.A gives an overview of the optimization described in the document and section III.B contains its analytical evaluation. Test results and graphics could be found in Section IV. The paper is terminated by a concluding Section V.

II. THE STATE OF THE ART

A. *Session Initiation Protocol Basics*

The Session Initiation Protocol (SIP)[4] is a protocol for establishing and tearing down multimedia sessions. SIP can also support various types of mobility such as terminal mobility, session mobility, personal mobility, and service mobility [5]. Since our focus in this paper is on terminal mobility we describe the SIP terminal mobility after briefly mentioning the SIP basics in the following subsection.

Fig. 1, shows a typical example of a SIP message exchange between two users, Alice and Bob. (Each message is labelled with the letter "M" and a number for reference by the text.) In this example, Alice uses a SIP application on her PC (referred to as a softphone) to call Bob on his SIP phone over the Internet. Also shown are two SIP proxy servers that act on behalf of Alice and Bob to facilitate the session establishment. This typical arrangement is often referred to as the "SIP trapezoid" as shown by the geometric shape of the dotted lines in the figure.

0-7803-8521-7/04/$20.00 (C) 2004 IEEE

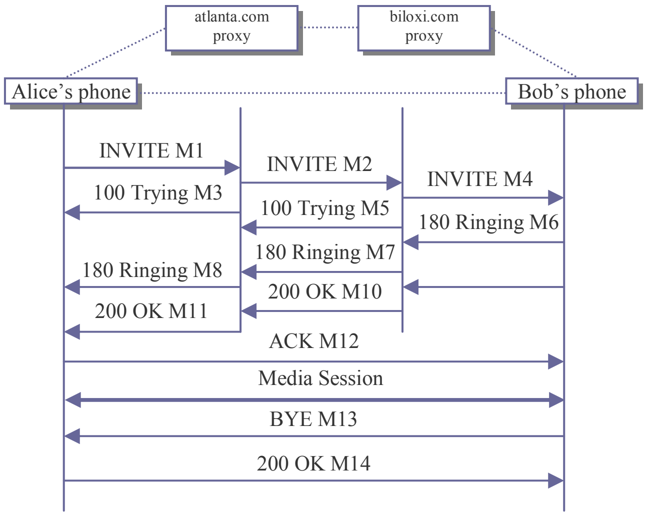

Figure 1. A basic SIP scenario.

Alice "calls" Bob using his SIP identity, a type of Uniform Resource Identifier (URI) called a SIP URI. It has a similar form to an email address, typically containing a username and a host name. In this case, it is sip:bob@biloxi.com, where biloxi.com is the domain of Bob's SIP service provider. Since Alice's softphone does not know the location of Bob or the SIP server in the biloxi.com domain, the softphone sends the INVITE request to the SIP server that serves Alice's domain, atlanta.com. The proxy server receives the INVITE request and sends a 100 (Trying) response back to Alice's softphone. The 100 (Trying) response indicates that the INVITE has been received and that the proxy is working on her behalf to route the INVITE to the destination. The atlanta.com proxy server locates the proxy server at biloxi.com and forwards, or proxies, the INVITE request there. The biloxi.com proxy server receives the INVITE and responds with a 100 (Trying) response back to the atlanta.com proxy server to indicate that it has received the INVITE and is processing the request. It then consults a database, generically called a location service that contains the current IP address of Bob and proxies the INVITE to Bob's SIP phone.

Bob's SIP phone receives the INVITE and alerts Bob for the incoming call from Alice so that Bob can decide whether to answer the call, that is, Bob's phone rings. Bob's SIP phone indicates this in a 180 (Ringing) response, which is routed back through the two proxies in the reverse direction. When Alice's softphone receives the 180 (Ringing) response, it passes this information to Alice, perhaps using an audio ring back tone or by displaying a message on Alice's screen.

In this example, Bob decides to answer the call. When he picks up the handset, his SIP phone sends a 200 (OK) response to indicate that the call has been answered. Finally, Alice's softphone sends an acknowledgement message, ACK, to Bob's SIP phone to confirm the reception of the final response (200 (OK)). The ACK is sent directly from Alice's softphone to Bob's SIP phone, bypassing the two proxies. This occurs because the endpoints have learned each other's address from the Contact header fields through the INVITE/200 (OK) exchange, which was not known when the initial INVITE was sent.

Alice and Bob's media session has now begun. During the session, either Alice or Bob may change the characteristics of the media session (e.g. media formats or endpoint location). This is accomplished by sending a re-INVITE containing a new media description. A re-INVITE scenario is discussed a bit later.

At the end of the call, Bob disconnects (hangs up) first and generates a BYE message. This BYE is routed directly to Alice's softphone, again bypassing the proxies. Alice confirms receipt of the BYE with a 200 (OK) response, which terminates the session.

B. Session Iinitiation Protocol Mobility

Reference [5] discussed how SIP can be used to support terminal mobility and its advantages over other mobility protocols. It is important to note that SIP-based terminal mobility does not add extra bytes to base SIP in order to support mobility. For the completeness of the paper, we briefly illustrate mid-call [5] mobility with Fig. 2a and Fig. 2b. Mid-call mobility allows a node to continue an ongoing session with a peer after changing networks.

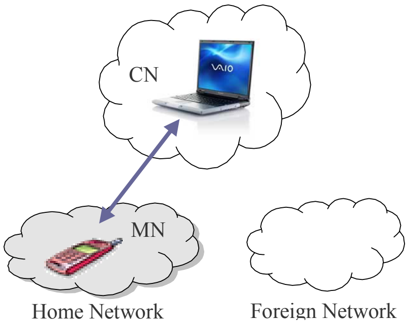

Figure 2a – SIP Mobility – before handover

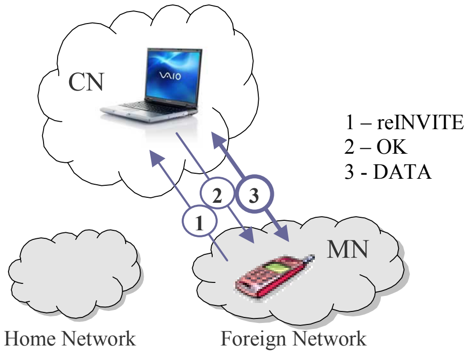

Figure 2b – SIP Mobility – after handover

Fig. 2, shows an example of how mid-call mobility is supported by SIP. In this example, an MN sends a re-INVITE request with its new IP address to the CN (1), and the CN directly sends packets to the MN at the new point of attachment to the network (2, 3). In this paper, we measure the handoff delay during mid-call mobility only.

C. *Mobility with Wireless LAN and IPv6*

IEEE 802.11b (WLAN) [6] enables two operational modes – ad hoc and infrastructure. When in ad hoc mode all nodes participate in the routing process. There are no key (central) nodes. An infrastructure mode is where nodes are in direct communication with a single key node called Access Point (AP). APs are dedicated equipment with at least one wireless and one wired interface. They serve as a bridge between wired and wireless networks. One or more nodes connected to an access point are called a Basic Service Set (BSS).

When an MN moves into a new BSS, it needs to synchronize with the corresponding AP. The synchronization procedure is initiated by the MN with the emission of a *Probe Request* frame to which an AP responds with a *Probe Response*. Once the synchronization completed, the MN starts an authentication procedure, and upon its successful completion, it moves to an association process where it receives BSS transmission parameters from the AP (e.g., the data rate and the transmission power). Once the association completes, the node can communicate via the new access point. This process, also known as L2 handover, is illustrated in Fig. 3.

When cover areas of different access points share a common cover zone, a node can roam between the access points. A node associates itself with the access point which offers the best signal or which has the minimum load among the access points.

On the network layer (L3) a Mobile Node (MN) detects movement between subnets by analyzing *Router Advertisements* sent by an Access Router (AR) [7]. An AR sends multicast *Router Advertisement* beacons at a random interval of between 0.03 to 0.07 seconds [8] for mobility aware networks and between 200 and 600 seconds [7] for standard network configurations. To determine whether network has changed the MN then compares router prefix information contained in these messages. If a *Router Advertisement* is not received within a specified interval, an MN may request one by sending a *Router Solicitation*. After Receiving the RA the MN creates its new network address. This is most often done using Stateless Address Autoconfiguration [9]. Once the address created, a host performs Duplicate Address Detection (DAD) [9] to make sure no other host on the local link is using the same address. DAD, however, is relatively expensive in terms of time. It consists in sending one or more *Neighbour Solicitations* using its new address and waiting for a response for at least a second. This considerably prolongs the handover stage. Therefore an MN should perform DAD in parallel with its communication (or not at all).

Once all this is completed an MN is ready to re-establish network activities interrupted during handover

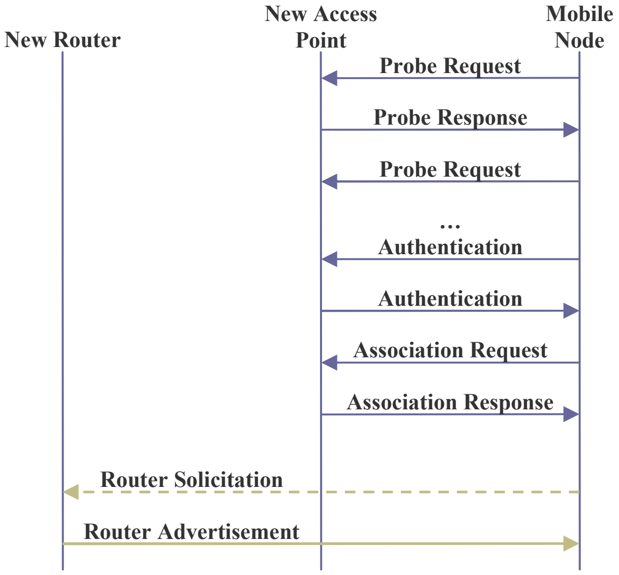

Figure 3 – the handover process

III. INTRODUCING L2 TRIGGERS IN SESSION INITIATION PROTOCOL MOBILITY

A. *Analytical Evaluation*

The handover process over WLAN, when controlled by an application layer entity, consists of 3 major phases – Layer 2 handover, Layer 3 address creation and Application Layer Handover which gives us a total handover time of :

$$T = T(L2) + T(L3) + T(APP)$$

1) T(L2) - Layer 2 handover - synchronization, authentication, and association. We will be calling the time necessary for a Layer 2 handover - T(L2)≈160ms [10].

2) T(L3) - Layer 3 address construction – *Router Solicitation*, and Router Advertisement. The time needed by Layer 3 to construct a new network address is referred to as *T(L3)*. T(L3)= T(movement detection) + T(DAD). Assuming that the movement detection process is terminated by the reception of an RA we have T(movement detection)≈750ms [7], and T(DAD) ≈1500s [9],[7]

3) T(APP) - Application Layer handover – Session re-initialization, referred to as T(APP) consists in the time that it takes the application layer to detect the address change, send the SIP re-INVITE request, and re-establish the media flow towards the new location of the MN. In our testbed this comes down to the following T(APP)≈T(app layer movement detection) + 50 ms

In this paper we focus on handover optimization by bringing T(L3) and T(APP) to a minimum. T(L3) depends mainly on the delay accumulated before receiving an RA. To bring that to a minimum an MN may send a Router Solicitation (RS) or a router may send RAs at a small interval. Both have their inconveniences: [7] specifies that responses to RSs must be randomly delayed by 0-500 ms and having dense RAs incurs extra traffic of approximately 14 kbps when sending RA frames at their minimum size of 88 bytes. On some networks, consumption of such bandwidth may be undesirable.

T(APP) may vary according to implementations. The only way for an application entity to get notified for a mobility event is to cache a node's network address and perform periodic verifications for a change. The exact interval at which these verifications would occur is quite delicate to determine since big values bring to a delay in detecting node movement and short intervals may take significant resources on the host machine.

In our solution a mobile node would send an RS right after moving to a new subnet. This would cause a nearby router to respond with an RA (after introducing an average delay of 250 ms) and speed up address construction by Layer 3. Once the MN receives the RA, it does not perform DAD in order to avoid the significant delay it implies, as explained in section 1.2. Right after that the application entity, controlling handover and mobility, would send its session re-initialization message to a CN.

Table I shows analytical time values needed by each layer to complete the handover process.

TABLE I. HANDOVER DURATION BY LAYERS

	SIP	SIP + DENSE RAs	SIP+CLT
L2 Handover – T(L2)	160 ms	160 ms	160 ms
L3 Addr Construction – T(L3)	1900 ms	50 ms	260 ms
SIP Handover	50 ms + detection	50 ms + detection	50 ms

B. *Optimization Description*

SIP is an application layer protocol and as such standard SIP entities could only rely on indirect mobility notification such as constantly scanning local addresses for a change. A mobility event would reach a SIP application only after all procedures described in section 1 have completed and, assuming that there are no lower layer mobility handlers (such as MIPv6 [8]), it is obliged to acquiesce to delays imposed by lower layers (e.g. waiting for an RA).

On the link layer level, though, there is an underlying awareness of connection events, which one might convey to a SIP application. If this information is available, it is always available faster than multicast-ed RAs and hence address changes.

Link-Layer movement, however doesn't necessarily mean that subnet and network address have changed so an RA must be solicited to confirm movement.

In our solution we introduce a Cross Layer Module (CLM), *Fig. 4,* that interacts with link, network, and application layer entities. Many network cards support notification model that enables communication with user-space processes. The CLM listens for event notifications and would send an RS every time an AP change is reported (i.e. right after the reception of an *Association Response*). The corresponding RA response is received and processed by the Network Layer. The network address is thus updated. The CLM on its turn uses the RA as a trigger to send a notification to a pre-registered application layer SIP entity and it is able to send its session re-initialization message (re-INVITE).

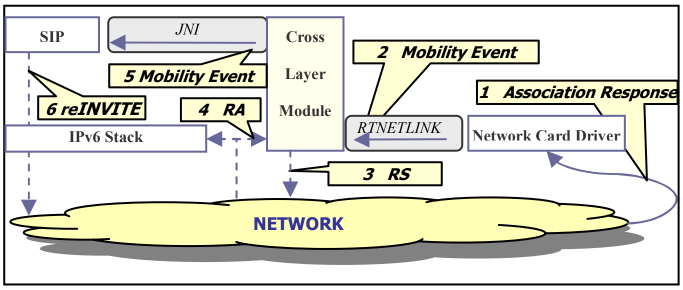

Figure 4 - Cross Layer Module Architecture

The implementation of CLM does not require any changes in the network configuration and will behave the same way over networks with dense RAs and standard RFC2461 [7] compliant networks (i.e. with density of RA emissions).

IV. IMPLEMENTATION AND TESTING

A. *Testbed*

The testbed used for testing the proposed optimization consists of three IPv6 access networks. Two of these are equipped with an 802.11b Wireless LAN access point. All APs are *Cisco Aironet 350 series*. Handovers occur back and forth the two IPv6 networks equipped with an AP. Network topology is shown on Fig. *5*.

MNs are Linux equipped terminals with modified IPv6 stacks so that no DAD is performed, as proposed by [7] and [11].

We implemented the CLM as described in the previous section. We also modified a SIP client called the SIP-Communicator [12] so that it would handle incoming re-INVITEs and interact with the CLM (i.e. send a re-INVITE upon CLM mobility notification).

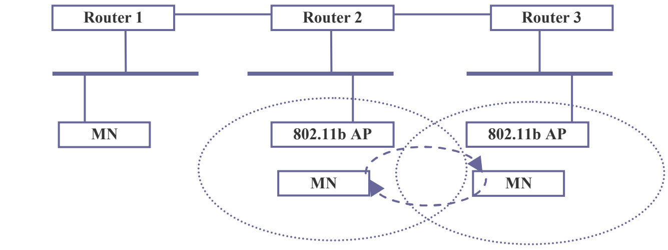

Figure 5 – Test bed network topology

B. *Test Results*

We have measured the handoff delay with SIP terminal mobility in our IPv6 testbed. Three different scenarios have been considered: (a) SIP mobility over a network with standard RA density (400 s) and with DAD; (b) SIP mobility over a network with dense RAs (750 ms) and with DAD; (c) Standard

SIP mobility over a network with dense RAs (750 ms) and without DAD; (d) SIP mobility of a node using Cross Layer triggers. The table shows the handoff delay for each of these configurations. Values shown in the table are the average result of ten successive handovers for each configuration.

TABLE II. HANDOVER LATENCY IN DIFFERENT SCENARIOS

Case	L2 start	L2 end	Address Config	reINVITE	OK	Data
(a)	0	0,165	407,54	409,149	409,192	409,2
(b)	0	0,165	0,755	2,149	2,199	2,247
(b)	0	0,168	0,666	0,699	0,756	0,797
(d)	0	0,167	0,313	0,346	0,395	0,454

Table II shows the signalling delay measured from the first Probe Request sent by the Mobile Node after entering the new BSS until the first UDP data packet received by the MN on its new location. As we can see, Layer 2 trigger usage combined with kernel modifications (removed DAD) have reduced handoff delay by 1500 ms to over 400 s for some cases. The same tests are illustrated on Fig. 6.

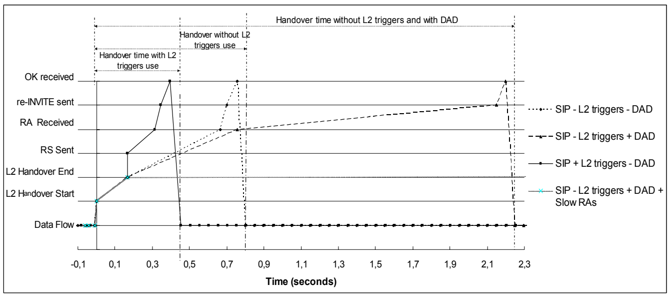

Figure 6 – Test bed network topology

V. CONCLUSION

In this paper, we have described an optimization of the handover process, its analytical evaluation, and sample results for SIP mobility in an IPv6 laboratory testbed. In our performance study we concentrate on optimizing the movement detection part of the handover process. We observe that using Cross Layer Triggers and removing DAD greatly improve SIP mobility. Results shown in the article reflect handoff delays for standard network configurations, networks with dense RA emissions, and MNs with and without kernel modifications and CLT usage.

REFERENCES

[1] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[2] S. Deering, R. Hinden. Internet Protocol, Version 6 (IPv6). RFC2460 – December 1998

[3] D. Johnson, C. Perkins, J. Arkko, Mobility Support in IPv6, RFC 3775. June 2004

[4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[5] H. Schulzrinne and E. Wedland, ACM SIGMOBILE Mobile Computing and Communications Review. Application-layer mobility using SIP - Mobile Computing and Communications Review, Volume 1, Number 2. February 2001.

[6] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ANSI/IEEE Std 802.11, 1999 Edition.

[7] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[8] D. Johnson, C. Perkins, J. Arkko, Mobility Support in IPv6, RFC 3775. June 2004

[9] S. Thomson, T. Narten, IPv6 Stateless Address Autoconfiguration. RFC2462. December 1998

[10] Nicolas Montavont, Thomas Noel. Analysis and Evaluation of Mobile IPv6 Handovers over Wireless LAN – 2001

[11] Nobuyasu Nakajima, Ashutosh Dutta, Subir Das, Henning Schulzrinne Handoff Delay Analysis and Measurement for SIP based mobility in IPv6. November 2002

[12] The SIP Communicator Project, url ref: http://sip-communicator.org

[13] Rajeev Koodli. Fast Handovers for Mobile IPv6 - Internet Draft - work in progress. draft-ietf-mipshop-fast mipv6-01.txt. 10 October 2003.

[14] G. Daley and JinHyeock Choi. Movement Detection Optimization in Mobile IPv6, Internet Draft - Work In Progress, draft-daley-mobileip-movedetect-01.txt. February 2003

[15] JinHyeock Choi, DongYun Shin. Fast Router Discovery with RA Caching in AP - Internet Draft – work in progress. draft-jinchoi-mobileip-frd-00.txt February 2003