Performance Evaluation of vertical handover mechanisms in IP networks

Andrea Polidoro, Stefano Salsano
DIE - Dept. Of Electronic Engineering
University of Rome "Tor Vergata"
Rome, Italy

Saverio Niccolini
Network Laboratories,
NEC Europe Ltd
Heidelberg, Germany

ABSTRACT

In this paper we discuss a methodology for performance evaluation of the solutions for vertical handovers between different wireless access technologies in IP network. The performance evaluation is based on simple analytical models and covers both the ideal case (no packet loss) and the real case where there is a given packet loss rate. The methodology is applied to a comparison among three solutions, namely MIPv4, classical SIP mobility management using re-INVITE messages and a SIP based solution called MMUSE (Mobility Management Using SIP Extensions) proposed by the authors.

I. INTRODUCTION

Let us consider a user owning a mobile device like a smart phone or a PDA. Mobility management solutions are needed to let the user access the services she needs while moving. The obvious user requirement is that no service disruption should be perceived and that no "manual" operations are needed to access the services while on the move, all the procedures should be automatic and produce a "seamless" experience for the user. Another obvious requirement is to consider the fact that a mobile device typically owns more that one interface (e.g. GPRS/UMTS, WiFi, Bluetooth…).

Focusing on IP based devices and services, these requirements for mobility management solutions can be tackled at various levels of the protocol stack from application level (e.g. SIP based solutions) to network level (e.g. Mobile IPv4, Mobile IPv6) to link layer level (e.g. 802.21). A large number of different mobility management solutions operating at these different levels have been proposed so far, both in the literature and in the standard bodies.

More often the solutions are proposed and discussed only at the architectural level, without an implementation or a performance analysis. Some works include an implementation and/or try also to make performance consideration and comparison among different solutions. In this work we will discuss first some methodological aspects about how the solutions should be evaluated and compared. We will try to identify the set of reference scenarios and the set of performance metrics that should be evaluated.

A solution for mobility management needs to include the procedures to keep track of device movements, while it does not necessarily need to provide support for handover of active sessions. However the handover capability is very important to provide a seamless user experience, and it is the most interesting benchmark for comparing different solutions. Therefore we will only consider solutions that provide the handover capability and we will compare these solution in their handover performance.

II. METHODOLOGY FOR PERFORMANCE EVALUATION

As stated earlier, the solutions for mobility management are often presented without discussing the performance related aspects and without providing results from implementation. When some implementation is discussed, it typically shows only some "success" scenario where handover signalling worked fine. We propose a methodology for comparing the performance of different handover solutions, including the proper evaluation of their behaviour in presence of packet loss events. In principle, the comparison could be performed with practical experiments (see for example [1], [2], [3]), with simulation or with analytical methods (see for example [4], [5]). We are interested in the analytical methods, and in particular we would like to extend the already proposed approaches by taking into account the loss rate that can affect the packets of the data and signalling flows involved in the mobility management procedures.

In the following subsections, we first classify the handover scenarios, then we define the performance metrics.

A. Handover Scenarios

With the help of Tab. 1, we classify the handover scenarios as follows. A device can have one single interface on a specific technology (e.g. WiFi) or more interfaces on the same or on different technologies (e.g. two WiFi interfaces, or one WiFi and a 3G interface).

	Handover types
One Single Interface	S1: "Sub-IP" S2: "Explicit HO"
More interfaces	M1: "Two-active-Ifs" M2: "Two-Ifs-one-breaks" M3: "One-active-IF"

Tab. 1 Classification of handover scenarios

In case of one single interface, we can have two cases which we name S1 ("Sub-IP") and S2 ("Explicit HO"). A "Sub-IP" handover is not perceived by the device at the IP level and above. For example WiFi handovers among access points in the same "ESS" (Extended Service Set) are fully handled at layer-2, so that the IP address of the terminal is not changed. Similarly a 3G interface will handle handovers in the cellular network across base station, without changing IP address. On the other hand the device may need to handle an "Explicit Handover" even on a single interface. This may be the case on a WiFi interface when going out from the coverage area of a set of access points operated by an organization and connecting to an access point operated by a different organization.

When (at least) two interfaces are involved, we envisage three different handover cases denoted as M1, M2 and M3. In the handover of type M1, the terminal is able to communicate over the two interfaces and the two interfaces remains active even during the handover execution. In the handover of type M2: the terminal is able to communicate over the two interfaces, but the communication on the "old" interface suddenly breaks, so that the communication needs only to be moved on the new interface. Finally, in the handover type M3 the terminal is connected on one interface both before and after the handover. In this case the communication on the new interface needs to be established "from scratch" as the first step in the handover execution. The three arrows in Fig. 1 represent the movement of a mobile terminal that performs the three different types of handover: in the M1 case the terminal is under double coverage of the two Access Networks (AN), in the M2 case the terminal is moving from an area of double coverage to an area of single coverage, in the M3 case the terminal moves from an area of coverage of AN1 to an area of coverage of AN2.

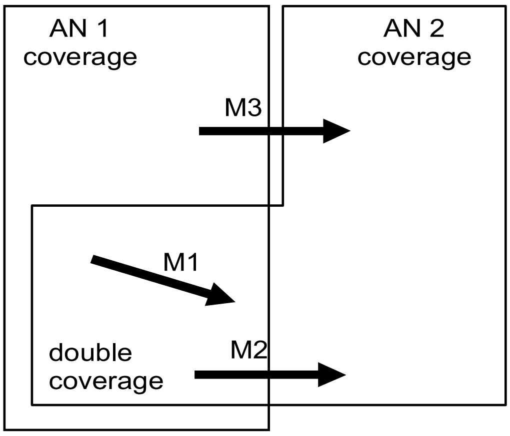

Fig. 1 Handover scenarios with 2 different access network technologies

B. Performance metrics

Our main performance metric is the "disruption time" dt perceived by the two users during the handover procedure. It can be different in the two directions, assuming that a mobile node is communicating with a fixed correspondent node. Therefore we define dt_{cn} as the disruption time for the uplink flows (from mobile node to correspondent node) as perceived by the correspondent node and dt_{mn} as the disruption time for the downlink flows (from correspondent node to mobile node) as perceived by the mobile node. When we want to evaluate a specific mechanisms for handover we will first classify the handover types as defined above and for each supported type Mi we will consider the sequence of messages to be exchanged. Then we can evaluate dt(Mi) as a function of the network delays among the involved entities and of the processing delays at the involved entities. In particular we define $\overline{dt(Mi)_{OK}}$ as the disruption time in case of a successful handover procedure (e.g. where no messages are lost) and $\overline{dt(Mi)_{FAU}}$ as the average disruption time assuming a loss probability for the message delivery.

The second considered performance metric is the number of mobility management signalling packets exchanged denoted as M. Similarly, we define $M(Mi)_{OK}$ as the number of exchanged signalling packets in case of a successful handover of type Mi and $\overline{M(Mi)_{FAU}}$ as the average number of exchanged signalling packets assuming a loss probability for message delivery.

We note that if we assign a weight w_i to the different handover types Mi, we could evaluate both the disruption time and the number of exchanged signalling packets as a weighted sum of dt(Mi) and M(Mi) respectively.

III. Mobility Management Mechanisms

In this section we introduce the mobility management mechanisms we compare in this paper: Mobile IP, SIP re-INVITE Method, SIP MMUSE. Due to space constraints it is not possible to provide a self contained description of these mechanism, therefore the interested reader is redirected to the suggested references.

Using Mobile IP, the mobile node can be reached with the same IP address, regardless of its movements. Mobile IPv4 (MIPv4) [6] foresees that packets incoming to the mobile node are sent to an entity called "Home Agent" (HA), tunnelled to the so called "Foreign Agent" (FA) and then delivered to the mobile node. Making an handover with "canonical" MIPv4 means moving from an "old" FA to a "new" FA and involves the HA as well. A lot of solutions have been proposed to improve the handover performance of Mobile IPv4

The SIP re-INVITE solution is the handover solution foreseen in SIP standards [7], [8]. It only relies on the capability of the mobile nodes. A terminal that is making the handover sends a request (a SIP re-INVITE message) to its correspondent, providing the new addresses for re-establishing the media flows.

The MMUSE (Mobility Management Using SIP Extension) handover solution has been proposed by the authors in [9] [10] [11]. It relies on an intermediate entity (the MMS, Mobility Management Server) to handle the handover. Under this solution, no support is needed from the correspondent terminal in order to perform the handover. The MMS can be seen as an extended Session Border Controller (SBC) [12]which is in charge of managing terminal mobility. Two drafts that describe the requirements and the solution itself have also been submitted to IETF: [13], [14].

IV. Evaluated Scenarios

The reference network scenario for our analysis is reported in Fig. 2. With reference to the handover scenarios described in section A above, we are going to analyse two of the cases with more interfaces: case M2 ("Two-IFs-one-breaks") and case M3 ("One-active-IF"). We define: t_h: as the delay between the Mobile Node (MN) and its Home Network (HN); t_s: as the delay between the MN and a Foreign Agent (nFA or oFA); t_{mc}: as the delay between the MN and the Correspondent Node (CN); t_{hc}: as the delay between the CN and the HN.

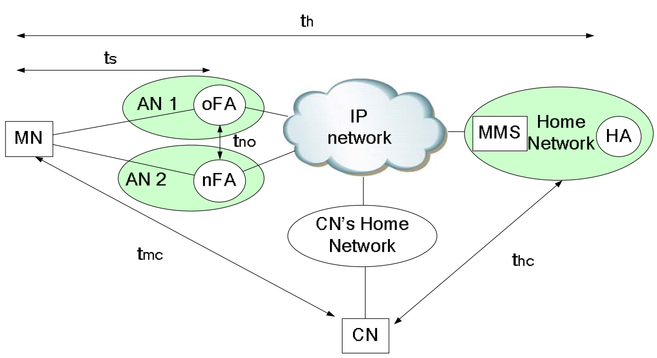

Fig. 2 The Reference Scenario

A. Case M2 ("Two-IFs-one-breaks")

In the MIPv4 case, we assume the mobile node already knows the new CoA offer from the nFA, so it only needs to activate the registration procedure. In the SIP cases, the MN has 2 IP addresses obtained with 2 different DHCP requests (one for each interface). The handover procedure is simply activated by sending a re-Invite (in the SIP re-INVITE solution) or a Handover Registration request (in the MMUSE solution). Fig. 3 shows the signalling flows of those procedures.

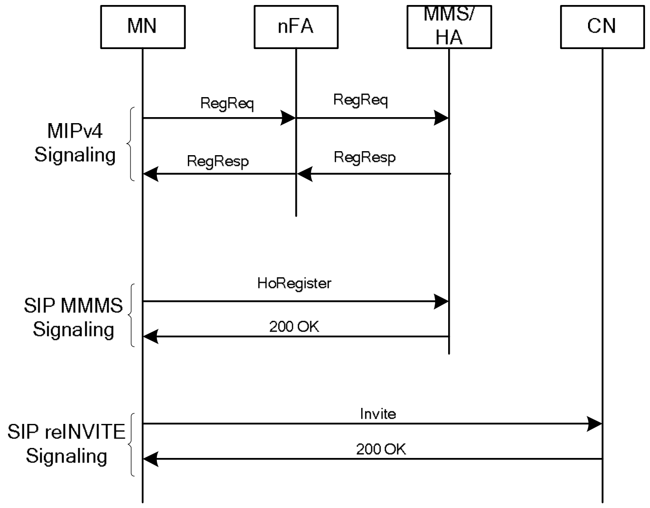

Fig. 3 Signaling Flows of Mobility Management Mechanisms in M2

B. Case M3 ("One-active-IF")

If the mobile node has only one interface with a valid IP address when this interface goes down, it must first acquire a new IP address on the new interface:

- in MIP [6] the MN must send a Proxy Rt solicitation to FA and waiting a response with his CoA. This procedure has a duration of 2 ts.

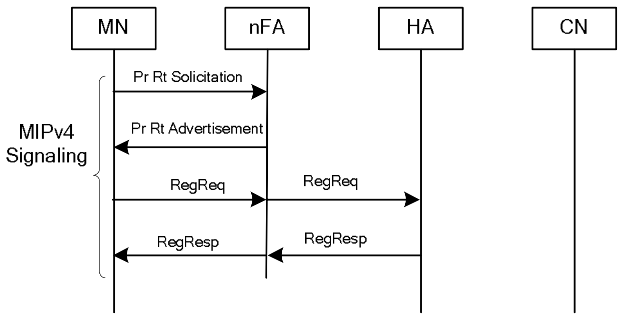

Fig. 4 MIPv4 Signalling Flows of Mobility Management Mechanisms in M3

- in SIP procedures we need another protocol (e.g. DHCP) to obtain an IP address, with a duration of 4 ts.

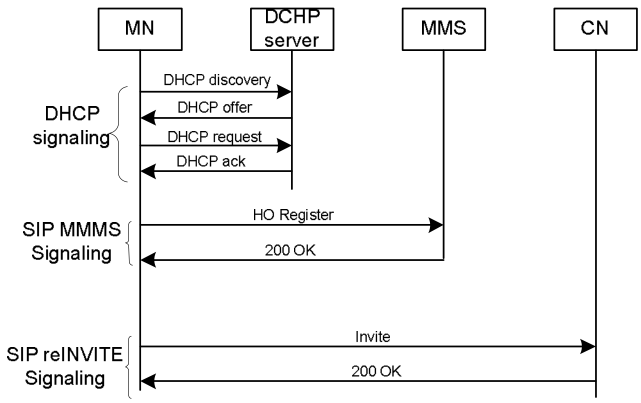

Fig. 5 SIP Signalling Flows of Mobility Management Mechanisms in M3

V. HANDOVER PERFORMANCE EVALUATION (NO-FAILURE CASE)

A. MIP

The MIPv4 procedure in the M2 case ("Two-IFs-one-breaks") is the following:

When the mobile node recognizes that the active interface has lost connectivity, it sends a Registration Request to the HA through the nFA. This message contains the CoA of the nFA (we suppose that MN has previously discovered this address).

After t_h time the HA receives this request and sends a Registration Response to MN. Simultaneously it forwards the media datagram received from CN to the MN. The MN receives the first media packet at:

$$dt_{MN} = 2t_h \tag{1}$$

after the break.

When the MN receives the Registration Response, it can send the first media packet through a new network, so the CN will receive this packet at:

$$dt_{CN} = 2t_h + t_{mc} \tag{2}$$

B. SIP re-INVITE

In the SIP re-INVITE solution, the MN sends a SIP re-INVITE message to CN, communicating the new IP address. When the CN receives this message (t_{mc}), it sends a 200 OK message and immediately sends the first media packet. So the MN receives the first media packets on the new interface at *2* t_{mc}. The CN stops receiving media packets through the old interface at t_{mc} after the start of the handover procedure, while it receives the first media packet through the new interface at *3* t_{mc} time. So the CN has a disruption time of *2* t_{mc}.

C. MMUSE-solution

In the MMUSE-solution, when the old interface breaks down, the MN sends the SIP HO-Reg and the media packets to the MMS (t_h) on the new interface. Therefore, the CN does not perceive any disruption.

When the MMS receives the Register, it sends the 200 OK and starts forwarding media packets that it receives from the MN to the CN and vice-versa. Note that the CN does not know the MN's status so it continues to send the media packets to MMS. The MN receives the first packet on the new interface at t_h after the old interface was lost.

In the case M3 ("One-active-IF") (case M3) we must add to all the disruption times the time needed to obtain a valid IP address on the new interface. This time is 2ts for MIP and 4ts for SIP procedure.

D. Comparison of handover Procedures

The following table shows the disruption time in the previously described scenarios.

	M2 case ("Two-IFs-one-breaks")	M3 case ("One-active-IF")
MIPv4	$\begin{cases} dt_{CN} = 2t_h + t_{mc} \\ dt_{MN} = 2t_h \end{cases}$	$\begin{cases} dt_{CN} = 2t_s + 2t_h + t_{mc} \\ dt_{MN} = 2t_s + 2t_h \end{cases}$
SIP re-Invite	$\begin{cases} dt_{CN} = 2t_{mc} \\ dt_{MN} = 2t_{mc} \end{cases}$	$\begin{cases} dt_{CN} = 4t_s + 2t_{mc} \\ dt_{MN} = 4t_s + 2t_{mc} \end{cases}$
MMUSE	$\begin{cases} dt_{CN} = 0 \\ dt_{MN} = 2t_h \end{cases}$	$\begin{cases} dt_{CN} = 4t_s \\ dt_{MN} = 4t_s + 2t_h \end{cases}$

Tab. 2 Disruption time in handover procedures

In order to compare the various handover mechanisms we plot the disruption time as function of the delay between MN and HN (Fig. 6) and as function of the delay between the MN and the CN (Fig. 7) in the M2 case. Fig. 8 and Fig. 9 represent the same analysis for the M3 case. In all the figures we have fixed the values: t_s=10ms t_{no}=5ms. Moreover in Fig. 6 and Fig. 7 we have t_{mc}=25 ms, while in Fig. 8 and Fig. 9 we have t_h=12 ms. This was same configuration as studied in [4] and [5]. Note that in each figures we are considering the maximum disruption time between dt_{MN} and dt_{CN} shown in Tab. 2

In Fig. 6 we notice that both MIPv4 and MMUSE grows linearly with the delay between MN and Home Network, with a slope of 2, but MIPv4 needs to add also the delay between MN and CN. The SIP-reINVITE solution is obviously not dependent on the delay between MN and HN, as it is only handled by the two involved terminals.

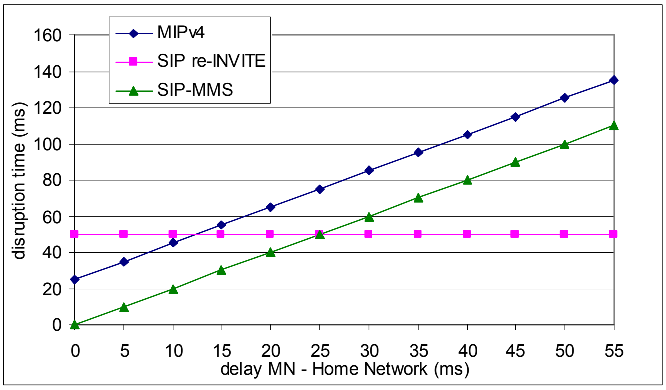

Fig. 6 Disruption as function of delay MN-HN in the M2 case

Fig. 7 shows the disruption time as function as to delay MN-CN. The MIPv4 and SIP-reINVITE solutions grow linearly with respect to this delay, but with a slope respectively 1 and 2. The MMUSE solution is not dependent on this delay and it has a constant value.

Fig. 8 and Fig. 9 show respectively the same situations in Fig. 6 and Fig. 7 but they add the delay necessary to discover the valid IP address for the new IF. So the slope of the disruption time is not varied but the all the delays are increased.

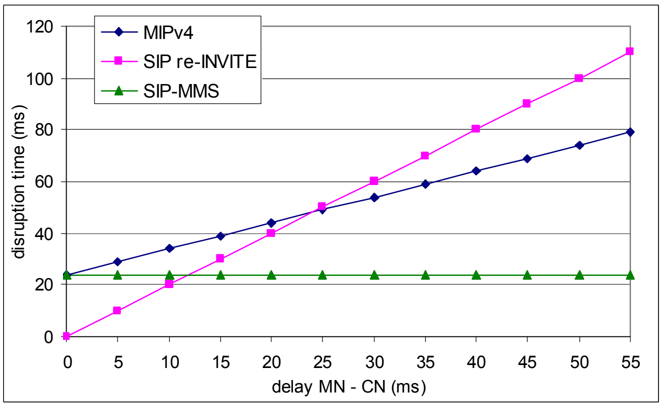

Fig. 7 Disruption time as function of delay MN-CN in the M2 case

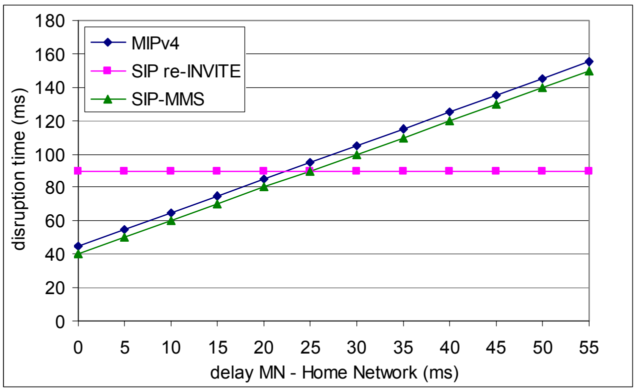

Fig. 8 Disruption time as function of delay MN-HN in M3 case

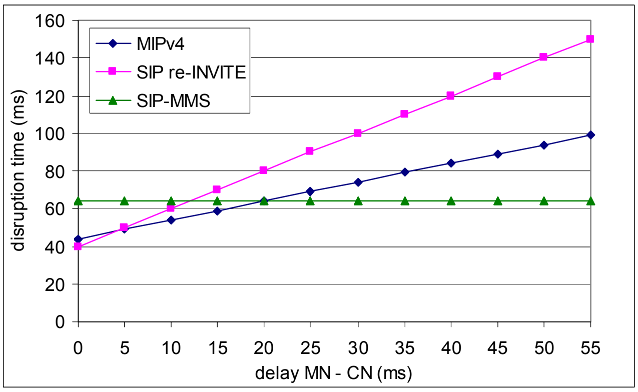

Fig. 9 Disruption time as function of delay MN-CN in M3 case

VI. Handover Performance Evaluation: Failure Cases

As for the performance analysis including failure cases, we only consider the M2 case for space constraint. We can have two types of failure case: 1) the "server-entity" does not receive a Request Message sent by the "client-entity"; 2) the "client-entity" does not receive the Reply Message sent by the "server-entity".

A. MIPv4

When a Registration Request or Registration Reply is lost, the MN waits for a timeout and then re-transmits the Request. The MIPv4 RFC [6] does not define a fixed value to this timeout, it define only a lower bound that is 1 s. We can consider this time as:

$$R^0_{mip}[ms] = 2RTT + 100 \geq 1000 \quad (3)$$

For each next retrasmission we must add twice this time, so if we define dt(0) as the disruption time evaluated in (2), we can have:

$$dt(n) = 2t_h + t_{mc} + R^0_{mip} \sum_{n=0}^{N} 2^n \tag{4}$$

B. SIP re-Invite

The SIP RFC [7] defines that the timeout at the first retransmission must be:

$$R^0_{sip} = T_1 \tag{5}$$

where T_1 is a value that estimates the RTT of the network ([7] suggests that T_1=*500 ms*). Two different retransmission mechanisms are defined. If we are retransmitting a Request for an "Invite transaction" the timeout is:

$$R_{inv} = \sum_{n=0}^{N} 2^n T_1 \leq 64T_1 \tag{6}$$

where n is the number of attempts. In case of a "Not-Invite Transaction" the timeout is:

$$R_{n\_inv} = \sum_{n=0}^{N} \min(2^n T_1, T_2) \leq 64T_1 \tag{7}$$

where T_2 should be set to T_2=*4s*

In the SIP re-INVITE solution, we must use the R_{inv} timeout, so we have this disruption time:

$$dt(n) = t_h + R^0_{sip} \sum_{n=0}^{7} 2^n \tag{8}$$

C. MMUSE solution

In the MMUSE solution, if the request is lost, we must send another request to MMS. Since a REGISTER message initiates a "not-invite transaction" the retransmission mechanism follows (7) so we have the following disruption time:

$$dt_{mms\_req}(n) = 2t_{mc} + R^0_{sip} \sum_{n=0}^{11} \min(2^n T_1, T_2) \tag{9}$$

The MMS starts sending media packets when it receives the REGISTER message. Even if the 200 OK response is lost, the handover procedure can be considered finished when the first media packet is received on the new interface. Therefore in this case the disruption time will become:

$$dt_{mms\_resp}(n) = t_h + n\Delta_{rtp} \tag{10}$$

where n means the number of RTP packets that are not received by the MN e Δ_{rtp} the inter-departure time of RTP packets.

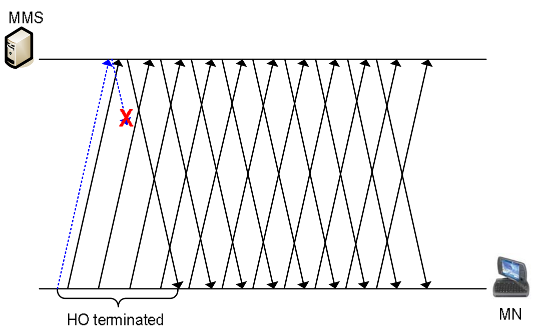

Fig. 10 MMUSE solution when the response are not received

In this way the information that the MMS has received the answer is not lost and due to the frequent transmission of RTP packets we are able to significantly reduce the disruption time, with respect to a traditional mechanism with retransmission of requests.

We note that the typical timing of request retransmission mechanism is not very well suited to handover of real-time connections, as the default retransmission timers are too long. Therefore we have proposed in our MMS based solution a faster retransmission mechanism ("MMS-fast") by changing the value of timers (T_1=50 ms, T_2= 200 ms). We note that this modification is compliant with SIP standards (which only suggest values for T_1 and T_2) and that it is very easy for us to apply this timer reduction only to handover register procedures. Of course the drawback of reducing retransmission timer is to increase the signalling load on the network and on servers due to "useless" retransmissions. We define a retransmission is "useless" if it is sent in the time before receiving the response to the initial request and there is no loss in the network. Fig. 11 shows the number of useless retransmission as a function of the RTT between Mobile Node and Home Network.

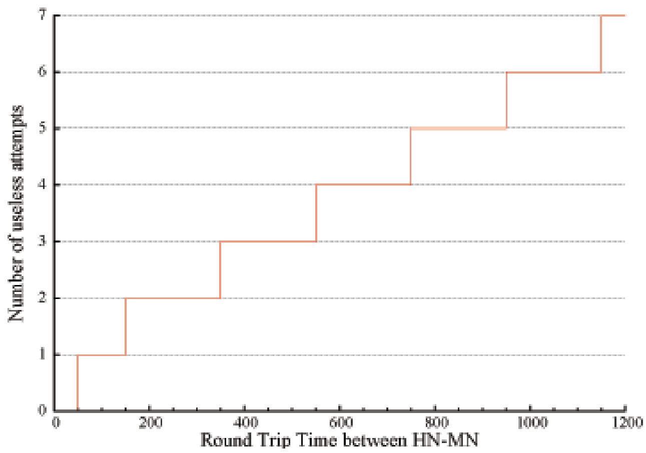

Fig. 11 Useless retransmissions vs. MN-HN RTT

Fig. 12 shows disruption time as a function of needed retransmissions. We can see that using fast-MMS the disruption time is less then 500 ms even if three retransmissions are needed.

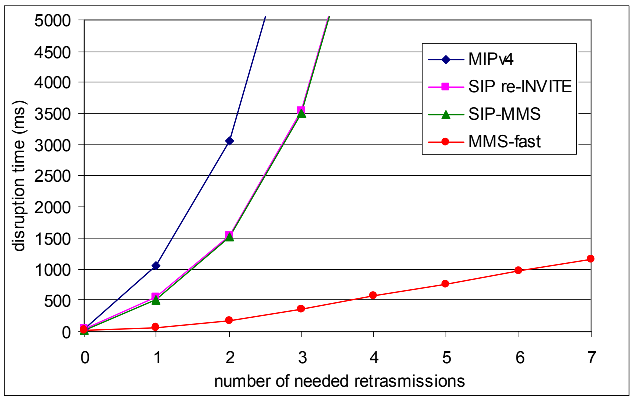

Fig. 12 Disruption time in the failure case

Let us p_h and p_{mc} be respectively the loss probability on the network paths MN-MMS and MN-CN. If we assume that p_h=p_{mc}=p we can approximate the average disruption time as:

$$\overline{dt} = \sum_{n=1}^{N_{max}-1} (\mathrm{dt}(n-1) \cdot p^n (1-p)) + \mathrm{dt}(N_{max}-1) p^{N_{max}} \tag{11}$$

where N_{max} is the Maximum number of retransmissions consents from the handover procedure.

Fig. 13 plots the average disruption time vs. the loss probability p when N_{max}=4.

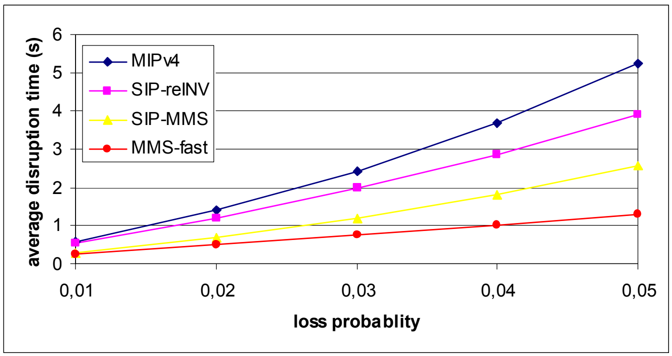

Fig. 13 average disruption time

VII. Conclusion

In this paper we have proposed a model to classify the mobility management solutions for vertical handover. The purpose of the model is to compare the handover solutions in term of disruption time both in the ideal case (without packet loss) and taking into account packet loss. The model has been used to compare the two reference solutions for mobility management at network lever (MIPv4) and application lever (SIP-reINVITE) with a second SIP based solution proposed by the author (MMUSE).

The analysis showed that the performances of MIPv4 and SIP-reINVITE mainly depends on the delay between the Mobile Node and the Correspondent Node, while the performance of MMUSE mainly depends of the delay between the Mobile node and the MMS node. If the location of the MMS can be controlled, MMUSE can easily outperform the other solutions.

The analysis confirmed that MMUSE performs better than other solutions also in presence of failures and retransmissions.

As future works we plan to address the performances of other mobility management mechanisms as MIPv4 low latency or MIPv6, with and without fast handover.

References

[1] Ashutosh Dutta, Sunil Madhani, Wai Chen, Onur Altintas, Henning Schulzrinne “Fast-handoff Schemes for Application Layer Mobility Management”, PIMRC 2004, Spain.

[2] A.Dutta, B.Kim, T.Zhang, S.Baba, K.Taniuchi, Y.Ohba, H.Schulzrinne “Experimental Analysis of Multi Interface Mobility Management with SIP and MIP”, IEEE Wirelesscom 2005.

[3] N. Banerjee, A. Acharya, S.K. Das, “Seamless SIP-Based Mobility for Multimedia Applications”, IEEE Network, March/April 2006

[4] T.T. Kwon, M. Gerla, S. Das, “Mobility management for VoIP service: Mobile IP vs. SIP” IEEE Wireless Communications, 10.1109/MWC.2002.1043856 (2002)

[5] H. Fathi, R. Prasad, S. Chakraborty, “Mobility management for VoIP in 3G systems: evaluation of low-latency handoff schemes” IEEE Wireless Communications, 10.1109/MWC.2005.1421933 (2005)

[6] C. Perkins et al. “Mobility support ofr IPv4” RFC 3344.

[7] J. Rosenberg et al., “SIP: Session Initiation Protocol,” IETF RFC 3261, June 2002.

[8] H. Schulzrinne and E. Wedlund, “Application-Layer Mobility Using SIP,” Mobile Comp. and Commun. Rev., vol. 4, no. 3, July 2000.

[9] S. Salsano, L. Veltri, G. Martiniello, A. Polidoro, "Seamless vertical handover of VoIP calls based on SIP Session Border Controllers" IEEE International Conference on Communications, ICC'06, 11-15 June 2006, Istanbul, Turkey

[10] S. Salsano, L. Veltri, A. Polidoro, A. Ordine “Architecture and testbed implementation of vertical handovers based on SIP Session Border Controllers”, Wireless Personal Communications, Springer, published on-line March 31, 2007, doi: 10.1007/s11277-007-9259-2 available at http://netgroup.uniroma2.it/Stefano_Salsano/papers/salsano-vo-sbc.pdf

[11] S. Salsano, C. Mingardi, S. Niccolini, A. Polidor, L. Veltri, “SIP-based Mobility Management in Next Generation Networks”, accepted for publication on IEEE Wireless Communication.

[12] J. Hautakorpi, Ed. “Requirements from SIP (Session Initiation Protocol) Session Border Control Deployments”, draft-ietf-sipping-sbc-funcs-04.txt, December 2007

[13] S. Niccolini, S. Salsano, L. Veltri “Requirements for vertical handover of multimedia sessions using SIP”, draft-niccolini-sipping-siphandover-02, Work in progress, February 2008

[14] S. Salsano, S. Niccolini, L. Veltri, A. Polidoro “A solution for vertical handover of multimedia sessions using SIP” draft-salsano-sipping-siphandover-solution-01, Work in progress, February 2008