A Generic Communication Architecture for End to End Mobility Management in the Internet

Lei Zhang, Patrick Sénac
ENSICA-LAAS/CNRS
Toulouse University, France
lzhang, senac@ensica.fr

Michel Diaz
LAAS-CNRS
Toulouse, France
diaz@laas.fr

Abstract

The proliferation of laptops, cellular phones, and other mobile computing platforms connected to the Internet has triggered numerous research works into mobile networking. The increasingly dense set of wireless access networks that can be potentially accessed by mobile users open the door to an era of pervasive computing. However, the puzzle of wireless access networks that tends to become the natural access networks to the Internet pushes legacy "wire-oriented" communication architectures to their limit. Indeed, there is a critical gap between the increasingly used stream centric multimedia applications and the incapacity of legacy communication stacks to insure the continuity of these multimedia sessions for mobile users. This paper proposes a generic communication architecture (i.e. not dedicated to a specific protocol or technology) that aims to fill the gap between the application layer continuity needs and the discontinuity of the communication service inherent to the physical layer of wireless mobile networks. This paper introduces an end to end communication architecture that preserves efficiently session continuity in the context of mobile and wireless networks. This architecture is mainly based on end to end mechanisms that could be integrated into a new generation reconfigurable transport protocol as defined in [20]. The proposed contribution efficiently satisfies mobility requirements such as efficient location management, fast handover, and continuous connection support.

Index Terms **— Mobility, location management, continuous connection, efficient handover.**

1. Introduction

Pervasive computing opens the way to a communication era where users get an ubiquitous wireless access to the Internet. Therefore, there is a paradigm shift from an end to end wired Internet to a wired Internet core accessed from ubiquitous wireless access networks. This paradigm shift has a sound impact on the legacy communication protocols that are not able anymore to efficiently support this evolution of the Internet. In this emerging pervasive computing era, computer networking will be carried on by a variety of mobile end-systems which can migrate anytime across different wireless subnets while keeping seamlessly their communication sessions. Nowadays, multimedia continuous streams (e.g. video or audio streams) take a bigger and bigger part of the information flows accessed or exchanged by Internet users. This feature enforces the requirement of offering seamless communication to mobile users. In today's internet, there is no widely disseminated, available and used communication architecture to efficiently and with a reduced infrastructure cost address the whole scope of mobility management issues. By performance of the mobility management, we refer mainly to 1) Continuous connection support**:** an established connection should be suspended instead of being cut off during the migration of the mobile nodes, and the continuous communication is available as soon as the host gets reconnected. 2) Fast handover: minimize the duration of handover to support a seamless communication. 3) Efficient location management: the current mobile node's network address should be accessible any time at a light cost.

In this paper, keeping the above three concerns in mind; we define a generic architecture for mobility management which involves both end to end and cross layer mechanisms. The rest of this paper is organized as follows. Section 2 discusses related work and the positioning of our architecture with respect to the state of the art. Section 3 introduces an analytical model that fixes the performance limits of the notion of continuous connection. Section 4 gives a detailed presentation of our architecture; the aim of our architecture is to offer a connection utility as close as possible to the analytical results given in section 3.

Section 5 introduces a first implementation of the proposed communication architecture for end to end mobility management. A conclusion will be finally given in section 6.

2. Related work

With the development of wireless communication technology, many protocols and mechanisms for mobile network have been proposed at different protocol layers. For instance, IEEE 802.11b, Mobile IP [2], MSOCKS [3] and SIP [22] focus respectively on the data link, network, transport and application layers. The question of the best suitable layer for mobility management has been recurrently raised. Several studies [4] have analysed the pro and cons of mobility management at these different layers. In [5], this question has been discussed in depth and evaluated by focusing on three important features that have to be addressed by mobile networks**:** seamless transition support, location management support and the changes in infrastructure. This study concludes that the transport layer is the strongest candidate to handle these mobility issues. Our solution is compliant with this approach. Indeed, the mobility is entailed by the end systems behavior. Following the "end to end hypothesis" that established the foundations of the internet, the communication architecture introduced in this paper contains the complexity of mobility management in the end systems. Such an end to end approach reduces network complexity and is much more able to deliver scalable solutions for the management of a very large population of end-systems. However, in today's complex networks, more and more evidences [15] show that one individual layer alone or the traditional interfaces between adjacent protocol layers like Transport/Network or Transport/Application can't provide sufficient information to allow transport mechanisms and applications to operate efficiently in a dynamic mobile Internet accessed from wireless networks. So, in order to make mobility behavior (disconnection, migration, unreachable node) more explicit to the upper layer, we need more elaborate cooperation schemes between upper and lower layers. Classic end-to-end mechanisms are not efficient enough for mobile host management because they are based on timescales which are in the order of magnitude of one or several RTTs they can't react quickly and accurately enough to the mobility behaviors and are exposed to the network QoS. Conversely, a vertical "cross layer" mechanism makes it possible for an end to end protocol to derive quickly end systems mobility status from local events delivered by the lower layers of the protocol stack.

Nowadays, most of the proposed mechanisms for mobility management address incompletely and partly the previously identified three issues that have to be tackled by architecture for mobility management. Atiquzzaman and Reaz's [6] classified in 4 categories the current transport layer mechanisms proposed for mobility management: (1) Handoff Protocol (like R2CP, MMSP and mSCTP [7]…), which can't be considered as complete mobility management schemes alone, but aim at improving performances such as latency and losses. (2) Connection Migration Protocol (like Freeze-TCP [8] and TCP-R [9]), which ignores handoff issues, but can efficiently manage the suspension of a transport connection during migration (handover). (3) Gateway based Mobility Schemes (like MSOCKS [3], ITCP [10], M-TCP [11], M-UDP [12] and BARWAN [13]), which require special entities that split the connection at the gateway between the MH and CN, but offer a partial solution to mobility management. (4) Mobility Management (Migrate [1] and SIGMA [14]), which provides relatively complete end to end mobility management schemes at the transport layer by implementing handoff and location management.

Our architecture keeps its roots in the migration connection scheme proposed in [1]. Moreover, we have integrated in our architecture the mechanisms of location management (DDNS [24] and HIP-Rendezvous mechanism [16]) to efficiently deliver continuous connections between two mobile terminals. In order to make explicit the mobile host's disconnection information to the upper layers, a "prediction disconnection" mechanism based on the data link layer is proposed.

Before describing our cross-layer architecture, we introduce a Markovian analysis of the notion of mobile connection. This analytical model gives the limit of the mobile connection utility at the physical layer and ignores handover processing overhead. This modeling gives the higher bound of the probability to be able to communicate for two corresponding mobile nodes. In practice, this limit can never be reached but can be approached thanks to our proposed architecture.

3. Mobile Connection Analytical Model

Differently from a classic transport connection, we define a mobile connection by a stochastic tuple**: (Source_Host Name, @S(t); Destination_Host Name, @D(t))**, where source and destination addresses

can dynamically evolve following the mobility behavior of the two end systems. Such a definition ensures a full separation between the host identity and its location. The dynamic association between host names and their current network address can be simply managed by DDNS [24] or the Host Identifiers protocol [17]. The above definition of a mobile connection leads to define a stochastic model that enables their performance analysis.

Based on a Markovian model, our analysis helps to estimate the stationary probability of simultaneous connectivity for the two corresponding mobile terminals in ideal conditions (i.e. lossless channel and zero delay handover). This estimated probability gives the higher bound of the mobile connection utility and can also be useful for mobility prediction purpose.

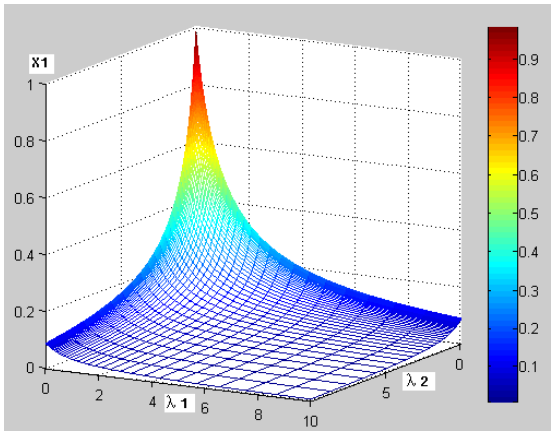

Fig 2 Probability of the simultaneous connectivity

We suppose that each of the two communicating mobile nodes MN_1 and MN_2 follows a model of mobility defined by a two-state continuous Markov chain where λ_1 (resp. λ_2) is the rate of the exponential distribution that defines for MN_1 (resp. MN_2) its probability to be able to communicate (i.e. to be attached to an access network) and μ_1 (resp. μ_2) the rate of the exponential distribution that gives for MN_1 (resp. MN_2) its probability to be disconnected from an access network. The global behavior of the two mobile nodes can be analytically modeled by composing these two continuous Markov chains.
Then from the resulting composed chain, the stationary probability of simultaneous connectivity for the two MHs (X1) can be derived and is given by:

$$X_1 = \frac{\mu_1 \mu_2}{\lambda_1 \lambda_2 + \lambda_1 \mu_2 + \mu_1 \lambda_2 + \mu_1 \mu_2} \quad (1)$$

This equation allows one to obtain the utility of a mobile connection as a function of the mobility behavior of the two mobile nodes. It is worth noting that the utility given by such a Markovian analysis considers the availability of the mobile connection only at the physical layer and does not take into account handover and higher protocols overhead; this is an optimal utility that an efficient architecture for mobility management should try to approach as close as possible but will never be able to reach.

For example, Figure 2 represents the probability of the simultaneous connectivity of the two mobile hosts when varying their respective mean residence time in their visited access networks (for fixed disconnection rates).

4. A new generation communication architecture for end to end mobility management

The analytical model introduced in the previous section defines the optimal connectivity delivered by the physical layer of two communicating MN. In the following we will progressively introduce an end to end solution for offering an efficient way to maintain logical connections even in presence of communication discontinuity entailed by the mobility behavior of the mobile nodes. Our architecture aims to introduce as few changes as possible in the current Internet protocol architecture. Indeed the mechanisms and protocols introduced in this section can be easily implemented as options of standard transport protocols or as components of a dynamically configurable protocol as defined in [20]. The proposed communication architecture comprises three principal parts: Mobility location management; Continuous connection support; Disconnection prediction method. Firstly, in the context of mobility as the two communicating hosts can potentially move anytime, a fixed point is necessary to conserve mobile nodes' updated location information (updated dynamic IP address for example). The Dynamic DNS and the HIP-Rendezvous-server [16] are the two approaches that we have experimented for our mobility location management. Secondly, during the migration of the mobile node, the on-going communication can be broken down because of sudden IP address changes; however, in such a situation a mobile connection is to be suspended instead of being cut off, and a protocol that allows a consistent reactivation of the communication has to be introduced. Thirdly, a mechanism for wireless disconnection prediction has been defined. Instead of detecting disconnections from time costly end to end mechanism, our proposed cross layer mechanism

anticipates this status directly from physical layer information inference.

4.1. Mobility location management

Two options for mobility location management have been studied and integrated in our architecture, Dynamic DNS and HIP Rendezvous server (RVS) mechanism. Dynamic DNS is more adapted to infrequently moving nodes, while RVS fits better with frequently moving nodes.

4.1.1. Dynamic DNS

Dynamic DNS allows domain names held by a name server to be updated dynamically. It allows mobile nodes' logical name (i.e. node's name + its original domain name) to be mapped with a varying dynamic IP address. Therefore, this makes it possible for other mobile nodes on the Internet to establish (or reestablish) connections to current mobile node without needing to track mobile node IP address themselves.

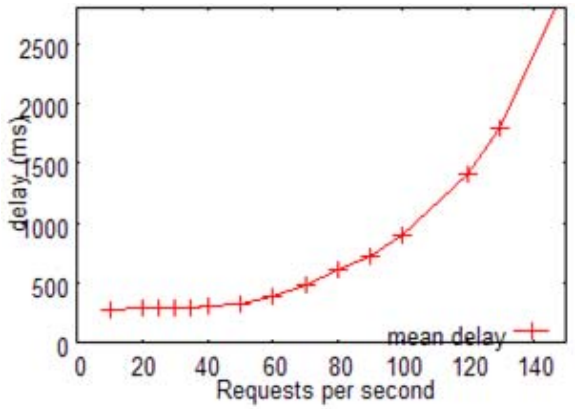

Fig 4 DDNS update delay according to load

In the DDNS option of our architecture, each mobile node has its own registered DNS server which offers the DDNS service. As soon as a mobile node acquires a new IP address in a new access network, it updates its IP address record in its own DNS server using TSIG [23]. Fig 4 shows the DDNS update latency in terms of the load of the DDNS server (i.e. the number of DDNS requests per second). This performance measurement shows that DDNS overhead is not negligible and that it is not an optimal choice when mobile nodes move frequently. Another shortage which can't be ignored is the security issue caused by the sharing of a secret key. However, we considered the use of this mechanism because it doesn't require any other additional infrastructure and doesn't change the current architecture of the Internet.

4.1.2 HIP Rendezvous server (RVS) mechanism

The Host Identity Protocol (HIP [17]) has been developed in the framework the IETF IP working group for a few years. HIP is a concrete proposal for adding a new secure and robust host namespace, managed by the Host Identity Protocol between the transport and the network layer. The new name space consists of *Host Identifiers*, which are cryptographic public keys robustly hashed to constant length Host Identity Tags (HIT) that can be more easily processed in protocol headers. HIP decouples host names from their network addresses which serve as pure locators. Instead of IP addresses, the applications and transport entities use *Host Identifiers* or Host Identity Tags to name peer hosts.

The HIP protocol also employs a new infrastructure, the HIP RendezVous Server (RVS) [16], which is used for efficient mobility location management purpose. The clients of an RVS are nodes that use the HIP Registration Protocol [18] to register their *HIT→IPaddress* mappings with the RVS. After this registration, other HIP nodes can initiate a base exchange using the IP address of the RVS instead of the current IP address of the node they attempt to contact. Essentially, the clients of an RVS become reachable at the RVS' IP addresses. Peers can initiate a HIP base exchange with the IP address of the RVS, which will relay this initial communication so that the base exchange may successfully complete.

For example, if a mobile node MN1 wants to establish a connection with MN2. Firstly, it obtains the IP address of MN2's rendezvous server from MN2's DNS record and then sends a HIP base exchange packet to this RVS. Then the RVS finds that the HIT contained in the arriving packet is not one of its own, so it checks its current registrations to determine if it needs to relay the packets. The RVS determines that the HIT belongs to MN2 and then relays this packet to the registered IP address of MN2. When MN2 receives the packet sent originally from MN1, it can then reply directly to MN1 without further assistance from RVS because the packet contains the new MN2's source address.

The HIP name space is cryptographic. Specifically, the host identity is a public key, and it signs a particular networking stack. By making the host identity a public key, authentication of protocol transactions is automatically enabled, and the protocol is robust to man-in-the-middle attacks. HIP has been

designed to be integrated with IPsec transport mode encryption (ESP). The HIP handshake for key establishment has also been designed to minimize the potential for denial-of-service attacks. We are currently implementing HIP and ail to measure its performance gain with respect to the DDNS approach.

4.2. Continuous connection support

In our architecture, an established network connection is suspended instead of being cut off if one or both of the mobile nodes migrate(s) and lose its on-going communication. [1] has introduced the notion of connection migration as an extended TCP option to address the continuity of a transport layer connection when only one of the two transport peers moves. In our architecture, with the help of an efficient mobility location service and the data link prediction mechanism to be introduced, we have extended the notion of connection migration (also called mobile connection) to encompass more general mobility scenarios where the two communicating peers can simultaneously move. Besides, for a transport protocol to be able to manage efficiently, he has to satisfy a minimum set of features among which connection management and message numbering are at the first rank. The robust and efficient management of mobile connections that supports the simultaneous migration of the transport peers raises several issues. Indeed, such complex mobility scenarios expose a protocol for mobile connection management to subtle potential cases of deadlock or address inconsistency. Therefore, the design of such a complex protocol cannot be addressed without a formal approach as explained in the next section.

Without entering into the details and complexity of our protocol for mobile connection management, we will just say that we extended the TCP standard state machine with new states and events that allow the management of mobile connexion to be done either from information explicitly sent by the peers' entities before moving or obtained from vertical cross layer interaction.

4.3. Disconnection prediction mechanism

The proposed mechanism is based on data link layer information; it makes our mobile node aware of its relative position in its current wireless access network and of its speed. The proposed location estimation technique makes it possible to predict when a mobile node will get disconnected from its current wireless access network according to the signal strength evolution.

For every wireless access point, a calibration based on a formula such as **SNR(dB) = A-B*log$_{10}$(distance)** can be initially done in order to establish the relationship between SNR (Signal Noise Ratio in dB) and the distance (in meter) from the access point of the current wireless access network (coefficients A and B vary according to the frequency of emitting signal) The basic idea of this estimation mechanism is to measure periodically the SNR (i.e. every ***n*** second(s)) so that we can calculate the relative speed V of the mobile node as**:**

$$V = [10^{(A-SNR[i])/B} - 10^{(A-SNR[i-1])/B}] / n ; \quad (2)$$

where SNR**[i]** is the current measurement and SNR[i-1] is the previous one). If V > 0, that means the mobile node is moving away from the AP, if V < 0, which means the mobile node is moving towards the AP. Note that this equivalent speed is not the real speed of the mobile node in the indoor case. For example, a sudden large speed variation corresponds to a brutal fall of the SNR, which can be caused by the breaking of obstacles between the mobile node and the access point. Therefore, the rough estimator must be enhanced with filtering techniques that aim to suppress the outlier.

If we define ***SNR_threshold*** as the critical threshold under which wireless communication cannot be supported anymore between the MN and the AP: then we can estimate the relative time (T) when the mobile node will get disconnected according to its current relative positive speed.

$$T = \frac{[10^{(A-SNR\_threshold)/B} - 10^{(A-SNR[i])/B}]}{V} \Big/ \frac{n * [10^{(A-SNR\_threshold)/B} - 10^{(A-SNR[i])/B}]}{10^{(A-SNR[i])/B} - 10^{(A-SNR[i-1])/B}} \quad (3)$$

However, the reality is somewhat different from this theoretical analysis. According to our experimental studies for an indoor case with obstacles wrapped up by Fig 5 (where 3 APs are located at positions 90, 210 and 380 respectively), we find that the measured SNR (in red line) is not always stably decreasing while the distance between the MN and the AP increases (that is, we observed a quite large variance of the measured SNR around the estimated one).

Therefore, these experimental results lead us to apply a low pass filter to the measured SNR in order to smooth its variation. This loss pass filter is based on an exponential moving average of the processed SNR given by the following formula:

$$SNR[i]^* = K1*SNR[i] + (1-K1)*SNR[i-1]^* \quad (0 < K1 < 1.0) \quad (4)$$

The so resulting SNR[i]* estimator is represented by the green line in Fig 5. Note that the choice of K1 has a

significant influence on the performance of our mechanism and is still under studies.

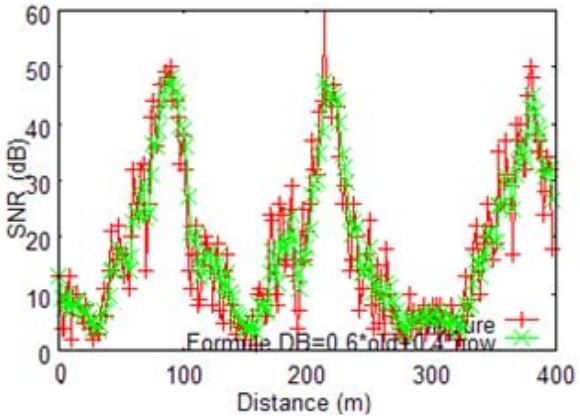

Fig 5 Signal strength evolution

In a first attempt we studied handover prediction mechanisms based on legacy protocols and mechanisms. We have assessed various predictors based on ICMP packets or message timeouts. In practice, these mechanisms induce feedback loops with a magnitude of several RTTs that can potentially entail lengthy handovers and discontinuities on multimedia streams. Moreover these network or transport layer mechanisms induce a waste of energy for a "blind" node that continues sending packets in vain while its corresponding node is disconnected or in conditions where the signal gets too weak to support the communication.

In our current architecture, a mobile node (MN1) can be immediately informed as soon as its corresponding node (MN2) is about to migrate. Such a "handover in progress" message avoids MN1 sending blindly while MN2 is out of connection. According to its auto-estimated relative speed and its estimation of the RTT between the two host, MN2 can automatically find a threshold (in terms of SNR) from which it starts to send informing packets to MN1 in order to assure with a given probability that MN1 will be informed of the handover to come.

This mechanism can also be integrated in the Pre-DHCP mechanism introduced hereafter. According to some experimental tests, we found that the time to acquire an IP address is often significantly higher for a wireless node than for a wired one. The main reason for this relatively high DHCP latency is that when the mobile node successfully associates with an AP, it may still be at the edge of the area covered by the AP. The strength of the signal may be very weak, so some link-layer frames corresponding to the DHCP request or its response may be lost. Figure 6 summarize the results of our measurements (for a D-Link DI774 wireless router) of the mean DHCP processing latency (ms) in function of the SNR(dB). These measurements show that at the edge of a wireless access network (i.e. near the SNR Cell Search Threshold=8dB in our experimental test), DHCP latency can be very high.

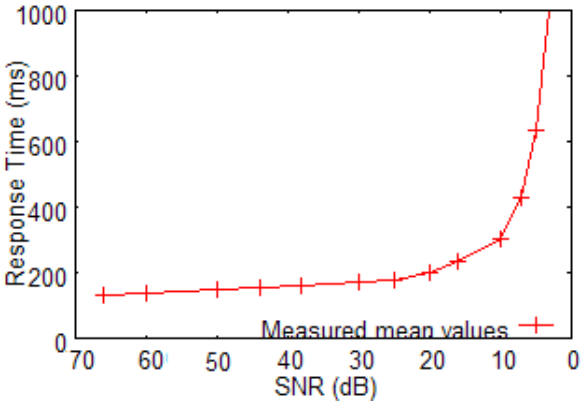

Fig 6 DHCP response time in terms of SNR

So there is a great interest in doing this DHCP processing before the handover, when the mobile node is still in its previous subnet, instead of doing it when the MN arrives at the edge of its new access network where the signal is weak. However, the moment when this pre-reservation of DHCP addresses in the previous access network starts is crucial, it should be triggered neither too early (i.e. IP addresses could be reserved in vain) nor too late (i.e. signal gets too weak or no enough time to finish the DHCP processing before getting disconnected). The disconnection prediction mechanism can help to find an optimal moment from which the mobile node starts its pre-DHCP processing via the current access router (a slight modification to Relay agents and the DHCP servers is needed to allow this pre-reservation mechanisms to be done) to acquire new IP addresses for the potential future wireless access networks to visit. The previously introduced handover estimation mechanism can guarantee that these DHCP pre-reservations will be successfully finished before the signal gets too poor or the current node gets disconnected. This significant improvement that makes possible soft handover gets rid of the DHCP latency. Note that this pre-reserved IP addresses are soft states associated with short lease duration.

This cross layer interaction that involves both the link, network and transport layers offers a generic solution that can be widely inserted in different protocols taking the benefit of obstacles detection, estimation of the moment when a mobile node will get

disconnected or estimation of how much time is left for a mobile node to have a good signal quality.

4.4. Global view of our architecture

Our generic architecture is mainly based on three important functions: location management, continuous connection support and explicit data link information processing / offering to upper layers.

The next description gives a big picture of the proposed generic architecture mobility management. Figure7 shows a generic scenario where the MN1 and MN2 have initially established a connection.

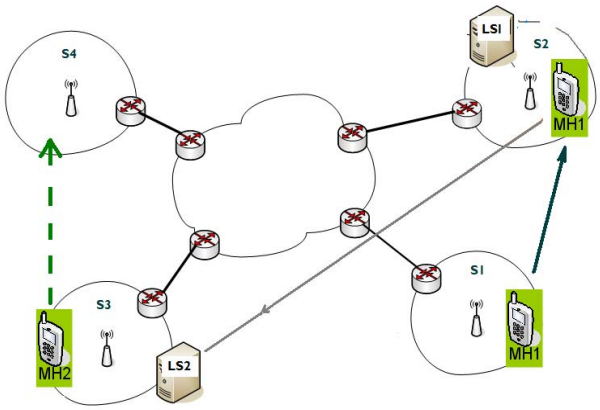

Fig 7 A simple mobility scenario

When MN1 starts to move away from subnet-1(S1), our data link prediction mechanism continuously estimates the optimal moment for starting pre-DHCP processing (There is a greater interest in getting rid of DHCP latency during handover in order to deliver seamless communication when there is a common coverage zone between the source and destination access networks.). It also guarantees with a given probability that the corresponding node MN2 is successfully informed by MN1's migration message before MN1 gets disconnected. Even if the communication between the two nodes is cut off because of MN1's migration, both of them conserve their current connection state. MN2 then enters into a ***WAIT*** state, and they note down both the sequence number of the last packet they respectively received. When MN1 arrives to its destination access network S2, it uses the pre-reserved IP address as its new IP address (if the lease duration of this pre- reserved IP address hasn't expired; if not, he should restart the DHCP processing). Then, MN1 saves its new IP address at once in its own location server LS1, where LS (Location Server) refers to a DDNS or a HIP-RV Server. In our mobile connection management protocol, a mobile node contacts its corresponding node as soon as it gets re-connected. MN1 then verifies MN2's current IP address in MN2's LS because MN2 might also migrate during MN1's migration. Then MN1 ***pro-actively*** sends a packet to MN2 to reactive the connection. However, if MN2 is migrating at this moment or the IP information in LS2 hasn't been updated on time, MN1 will enter into a ***WAIT*** state and wait being woken up by MN2 until MN2 finishes its migration and gets ready for communication. When the communication is recovered, they resume their communication from the last packet they respectively received.

In this section, we have introduced a simple and generic scenario that involves the main components of the proposed architecture for mobility management. This scenario has been modeled and validated in the TURTLE formal language [21], an integration of Real Time Lotos and UML. TURTLE is a UML profile dedicated to the modelling and formal validation of real-time systems. Fig 8 shows one example of the simulation results obtained from TTool. In this simulation scenario the two mobile nodes respectively migrate during the time intervals [150,250], [400,500] and [260,360], [620,650].

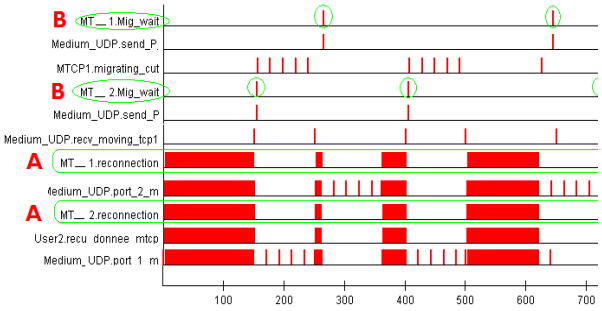

Fig 8 A T-Tool simulation run

Lines B represents the moment the nodes enter into the ***WAIT*** state. Lines A represent the time when mobile nodes are able to communicate. The formal modelling of our communication architecture allowed us to exhibit initial design choice that induced potential deadlocks and inconsistencies and to finally insure the liveness and consistency of our protocols.

5. A first rough implementation

This section introduces a first rough implementation of our generic architecture. This scenario has been implemented in JAVA under the Linux 2.2 kernel (i.e. Ubuntu) and in user space above UDP. We used D-link: DI-774 as wireless routers. We supposed a 95dB

constant noise level and a **8dB** critical communication Threshold. We experimentally found that the coefficients A and B, associated to the formula which describes the SNR (dB) in terms of distance for the DI-774 wireless routers, are respectively A=60 and B=20.5, that is:

SNR (db) = 60-20.5* $\log_{10}$(distance) **(5)**

In this first implementation, we just added sequence numbers to the UDP payload and intentionally didn't address rate, error and congestion control issues. We implemented the DDNS option for the location management. We used the java package dnsjava (V.2.0.3) [19] to update the dynamic IP address in MN1's original DNS server; a domain name in this DNS is always associated to MN1's dynamical IP address.This implementation has been tested in an indoor environment with a great diversity of potential obstacles between the MN and the access points. Figure 9 represents this scenario, where a video file is transferred from a mobile node (MN1), which moves across different APs, to its corresponding node (MN2).

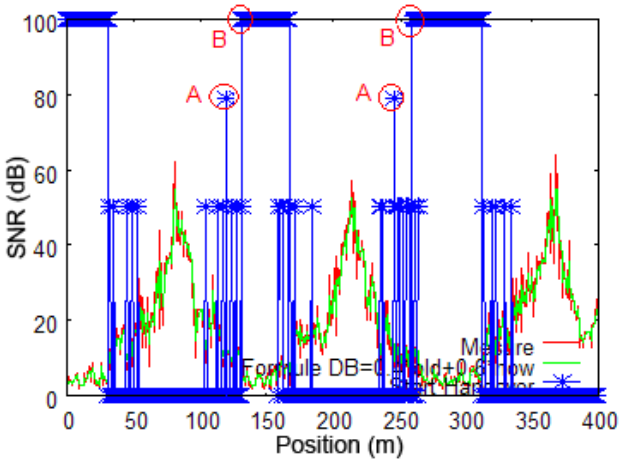

Fig 9 Signal strength evolution along a MN trajectory

Fig 9 gives the signal strength evolution during this mobility scenario. In this scenario MN1 moves successively through three Wireless Access Networks of which the respective APs are at the relative positions 80, 215, and 370 along MN1's trajectory. In Fig.9, the red curve gives the measured SNR; the green curve gives the value of the filtered SNR used for the prediction calculation. The points labeled by A represent the moments when pre-DHCP processing is started at relative positions 118 and 247 which correspond respectively to SNRs of 10.9dB and 11.5dB. Points B (positions 121 and 251) represent the moments when MN1 starts activating the end to end protocol for sending packets in order to inform its corresponding node that MN1 starts a handover. According to the experimental results, the proposed prediction mechanism allows MN2 to be informed on time of MN1's migration.

Figure 10 shows the evolution of the sequence number of the packets sent by MN1 and received by MN2 around the first handover. Note that the disconnection duration 15s corresponds to the relative interval (130-165) of MN1's trajectory which in this case is supposed going to a walking speed. With our proposed prediction mechanism, we can see that the sender stops sending packets just before he gets disconnected during a handover and that there is no loss entailed by the handover and none packet uselessly sent.

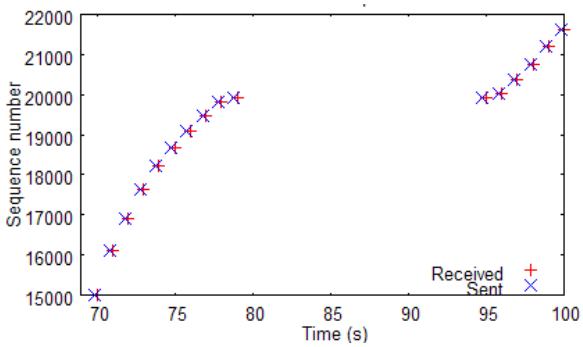

Fig 10 Evolution of the message sequence numbers

This first implementation doesn't focus on error, congestion control and security issues. It insures the consistent continuity of stream delivery, after a handover, thanks to a simple buffering by the receiver of the last received sequence number.

This first implementation will also progressively integrates the results of our current studies on congestion and error control mechanisms adapted to mobile systems in the Internet.

6. Conclusion

This article has introduced a generic framework for the efficient management of mobile end systems in the Internet. This framework aims to respect the end to end hypothesis that founded the design of the Internet. Indeed, our contribution contains the complexity of mobility management in the end systems, minimizes its impact on legacy protocols and introduces a minimal change to the current Internet architecture by just introducing few application layer services for dynamic address resolution or Quality of Service control. We argue that such an end to end approach is a strong

candidate for allowing scalable and high performing deployment of mobility management in the new generation Internet. Our current research work focuses on introducing in this framework coherent congestion control and QoS control mechanisms that aim to apply jointly an efficient adaptation between the potentially highly dynamic network status and the not less dynamic users' behavior and applications' needs.

7. References

[1] An End-to-End Approach to Host Mobility. Alex C. Snoeren and Hari Balakrishnan MIT Laboratory for Computer Science 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)

[2] C. Perkins, .IP mobility support.. IETF RFC 3344, August 2002.

[3] D. A. Maltz and P. Bhagwat, MSOCKS: An architecture for transport layer mobility, IEEE INFOCOM, San Francisco, CA, pp. 1037-1045, March 29 - April 2, 1998.

[4] T. R. Henderson, Host mobility for IP networks: a comparison," IEEE Network, vol. 17, no. 6, pp. 18-26, Nov - Dec 2003.

[5] W. M. Eddy, At what layer does mobility belong? IEEE Communications Magazine, vol. 42, no. 10, pp. 155 - 159, October 2004.

[6] Atiquzzaman and Reaz's. Survey and Classification of Transport Layer Mobility Management Schemes. INVITED PAPER

[7] S.J. Koh, M. J. Chang, and M. Lee, mSCTP for soft handover in transport layer, IEEE Communications Letters, vol. 8, no. 3, pp. 189-191, March 2004.

[8] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, Freeze-TCP: a true end-to-end TCP enhancement mechanism for mobile environments, IEEE INFOCOM, Tel Aviv, Israel, pp. 1537 - 1545, March 26-30, 2000.

[9] D. Funato, K. Yasuda, and H. Tokuda, TCP-R: TCP mobility support for continuous operation, IEEE International Conference on Network Protocols, Atlanta, GA, pp. 229-236, October 28 - 31, 1997.

[10] A. Bakre and B. R. Badrinath, I-TCP: indirect TCP for mobile hosts, IEEE International Conference on Distributed Computing Systems, Vancouver, Canada, pp. 136 -143, May 30 - June 2, 1995.

[11] Z. J. Haas and P. Agrawal, Mobile-TCP: an asymmetric transport protocol design for mobile systems, IEEE ICC, Montreal, Canada, pp. 1054 - 1058, June 8 - 12, 1997.

[12] K. Brown and S. Singh, "M-UDP: UDP for mobile cellular networks, Computer Communication Review, vol. 26, no. 5, pp. 60 - 78, October 1996.

[13] R. H. Katz, E. A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G. T. Nguyen, V. Padmanabhan, and M. Stemm, The bay area research wireless access network (BARWAN), IEEE COMPCON, Santa Clara, CA, pp. 15 - 20, February 25-28, 1996.

[14] S. Fu, L. Ma, M. Atiquzzaman, and Y. Lee, "Architecture and performance of SIGMA: A seamless handover scheme for data networks, IEEE ICC, Seoul, South Korea, May 16-20, 2005.

[15] Lars Eggert, Wesley M. Eddy. Towards More Expressive Transport-Layer Interfaces. MobiArch '06 San Francisco, California, USA.

[16] J. Laganier, L. Eggert. Host Identity Protocol (HIP) Rendezvous Extension. Draft-ietf-hip-rvs-05

[17] R. Moskowitz, P. Nikander. Host Identity Protocol Architecture. Draft-ietf-hip-arch-03

[18] J. Laganier , L. Eggert, T. Koponen. Host Identity Protocol (HIP) Registration Extension. Draft-ietf-hip-registration-02

[19] Dns java. Version 2.0.3. http://www.dnsjava.org/

[20] E. Exposito, P. Senac, Michel Diaz, Compositional Architecture Pattern for QoS-oriented Communication Mechanisms, Multimedia Modelling 2005, Melbourne, Jan. 2005

[21] TURTLE : http://labsoc.comelec.enst.fr/turtle/ttool.html

[22] J. Rosenberg. SIP: Session Initiation Protocol. RFC 3261

[23] D. Eastlake. DNS Request and Transaction Signatures. RFC 2931

[24] P. Vixie, Editor, Dynamic Updates in the Domain Name System. RFC 2136