

Taxonomy and analysis of IP micro-mobility protocols in single and simultaneous movements scenarios

G. De Marco^{a,*}, L. Barolli^a and Salvatore Loreto^b

^a*Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT), 3-30-1 Wajiro-Higashi-ku, Fukuoka 811-0295, Japan*

^b*Dipartimento di Informatica e Sistemistica, University of Naples “Federico II”, via Claudio 21, 80125, Napoli, Italy*

Abstract. The micro-mobility is an important aspect in mobile communications, where the applications are anywhere and used anytime. One of the problems of micro-mobility is the hand-off latency. In this paper, we analyse two solutions for IP micro-mobility by means of a general taxonomy. The first one is based on the Stream Control Transmission Protocol (SCTP), which allows the dynamic address configuration of an association. The second one is based on the Session Initiation Protocol (SIP), which is the most popular protocol for multimedia communications over IP networks. We show that for the SCTP solution, there is room for further optimisations of the hand-off latency by adding slight changes to the protocol. However, as full end-to-end solution, SCTP is not able to handle simultaneous movement of hosts, whose probability in general cannot be neglected. On the other hand, the SIP can handle both single and simultaneous movements cases, although the hand-off latency can increase with respect to the SCTP solution. We show that for a correct and fast hand-off, the SIP server should be statefull.

Keywords: Mobile IP, wireless networks, SCTP, SIP, simultaneous movements

1. Introduction

Most of the expectations of next generation Internet applications are around wireless applications which will enable the ubiquitous communications. These kind of applications require the extension of IP connectivity over the radio medium. Standardised systems like GSM/GPRS in Europe and WCDMA in Japan can offer Internet access by handling all the micro-mobility difficulties at Link Layer (LL or L2). This is possible because of a sophisticated network infrastructure. With micro-mobility we mean the movement of nodes within a subnetwork. On the other hand, the widespread deployment of WLAN led to envision cheaper solutions for wireless Internet, which in turn lack of a robust and centralised mobility management. The mobility management in WLAN introduces several problems, especially for small size cells where the rate of hand-off can be high. For example, one of the problems of the IEEE 802.11 WLAN technology is the high latency of an inter-Access Point (AP) hand-off, that is the switching or the association from one AP to another. Early measurements have shown that this kind of L2 hand-off roughly lasts 500 ms [9]. The main factor which affects the L2 hand-off delay is the

*Corresponding author. Tel.: +81 92 606 4970; E-mail: demarco@fit.ac.jp.

channel scan phase, that is the searching for a new radio channel own by a neighbouring AP. Recently, a proposal to speed up the channel scan phase of the IEEE802.11 has shown to reduce link layer latencies to tens of milliseconds. However, the situation worsens if one counts also the latencies associated to higher layer hand-off, for instance the network layer (L3) hand-off. Traditionally, the Mobile IP Protocol (MIP) has been proposed as L3 protocol to provide Mobile Nodes (MNs) with the ability to maintain the IP address anywhere the MN is located [6]. In this case, the mobility is transparent for applications running on top of MIP. Another solution to handle mobility is that based on Session Initiation Protocol (SIP) [5,11]. SIP is an application protocol and has been basically designed for IP telephony signalling. It provides the classical concepts of localisation servers to retrieve the current and visible address of the MN, as MIP does. If SIP is used as mobility management, all other applications, like Web and FTP, should be modified in order to use the SIP functionalities. Both in SIP and MIP, the network must employ dedicated servers: MIP Home Agents (HA) and Foreign Agents (FA) and/or SIP servers.¹ In order to reduce delays and packet losses due to the hand-off process, a lot of enhancements have been proposed for MIP. An example is the Hierarchical MIP (H-MIP) and its variants. In general, in all the proposals, the network should be augmented with additional components, like Mobility Anchor Point in MIP, or MIP Location Registrar (MIP-LR). These proposals are far from the end-to-end approach of the current Internet design. In this paper, we compare two solutions to the micro-mobility, with minimum changes to the core network. The first one is based on the Stream Control Transmission Protocol (SCTP) for in-session mobility management. The second one is based on SIP. Both solutions require changes to the base protocol in order to decrease the hand-off latency. Moreover, in the case of simultaneous movements, a location or *rendez-vous* server is essential. Hence, only SIP can be used, by guaranteeing the routing of the signalling messages through a statefull server. To the best knowledge of the authors, this is the first attempt to evaluate whether SIP and SCTP can be jointly used for mobility, without introducing neither additional components in the network nor redundant operations, as we discuss in the following. In Section 2, we show different types of architecture for micro-mobility by means of a general taxonomy of mobility functions. In Section 2.3, we focus our attention on SCTP and SIP, by analysing their features both in the single and simultaneous movement case. In Section 3, we first analyse the latency caused by hand-off process for the case of the reliable transmission with SCTP; then, in Section 3.2, we analyse the performance of our proposed optimisation and show a brief comparison with SIP along with possible optimisations. Conclusions are given in Section 4.

2. Network architectures for mobility

In IP micro-mobility the following functions are fundamental:

- The identification function, $f : x\text{-Id} \rightarrow x\text{-loc}$, which is used when an originating entity, e.g. a Corresponding Node (CN) or another MN, wishes to communicate with the MN (data are transmitted to or received from it). The originating entity knows the identity of the destination, $x\text{-Id}$ (e.g. $x\text{-Id} = \langle \text{giuseppe}@sip.fit.ac.jp \rangle$), and needs its locator, $x\text{-loc}$.
- The localisation and forwarding function, $g : x\text{-loc} \leftrightarrow \text{IP}$, which maps the x 's locator to a visible IP address and viceversa. The g -function is invoked in order to localise the destination IP address, which can vary over time, e.g. whenever an L3 movement takes place in the case of MN travelling

¹SIP server can be: Registrar Server, Proxy Server, and so on. We use the general word SIP server by assuming that a dedicated function is inserted in the SIP server whenever it is required.

among different subnetworks. In general, $g(\cdot)$ is not injective, that is g can have multiple IP addresses associated to its locator.

- The updating function, $u: (x\text{-loc}, IP_{old}) \rightarrow (x\text{-loc}, IP_{new})$, which is usually executed by MN when its IP address has to be changed. The function will also update the MN's IP_{old} address at the locator agent, e.g. the SIP server.

For example, in MIP, $f(x\text{-Id}) = IP_{Home}$, that is the *caller* knows the Home IP address of the destination.² Then, the HA performs $g(IP_{Home}) = MN_{CoA}$, that is the HA will forward packets towards the current Care of Address (CoA), MN_{CoA} , of the MN. Inside the MN, a Web-like application will first contact a DNS to get the IP address of the Web site. If the Web site is not moving, $f(x\text{-Id}) = IP_{Web\text{-site}}$ and g function is not needed. On the other hand, if the MN will move, it must (re-)run u in order to update its IP at the HA and, if Routing Optimization (RO) is available, it should also update the cache of the CN. We note that, if the Web site server is not MIP-compliant, the RO is not available and packets will be always sent to the MN's HA, that is the Web server will need g function during communication. In this case, communication delays are not optimised (triangular routing problem).

The same happens in SIP, mostly used for VoIP applications. The function f is performed by the SIP-*caller* whenever it wants to communicate with another SIP entity: a SIP INVITE message is sent to the SIP server of the *caller*. The g function is then called inside the SIP server which will provide the SIP-*caller* with the current IP address of the *callee*. In this case, $g(x\text{-loc}) = IP_{MN}$; if the SIP-*callee* belongs to a different SIP domain, $g(x\text{-loc}) = IP_{Sip\text{-callee}}$, that is the IP address of the SIP server which the callee is registered with. If the MN gets a new IP address, the MN will perform u function, by updating the information in the SIP server, as shown in Fig. 1-a in the case of a pre-session mobility. The SIP server will execute the g function, as above. For in-session mobility (Fig. 1-b), usually, this operation requires a Re-INVITE message from the MN, i.e. $u(x\text{-loc}, IP_{old}) = (x\text{-loc}, IP_{nCoA})$. In the next sections, we see that if SIP server does not take any role in this operation, the protocol is not correct. Let us note that in this case the forwarding of packets is accomplished by the standard IP protocol, and the network must obviously comprises SIP servers.

There are also other proposals which try to handle both single and simultaneous movements of MN at the application layer. For example, [3] propose to use a dedicated server in a particular domain in order to overcome the misbehaviour in the case of simultaneous movements (see following sections). Although the work in [3] was aimed at the management of mobility of MN across heterogeneous subnetworks (read vertical hand-off), the idea of the dedicated server is indeed similar to that of the standard SIP Proxy.

SCTP can be used for the u function without the support of dedicated location servers as the HA and FA in MIP. In fact, SCTP can dynamically add multiple IP addresses during the communication, or association in SCTP terminology, by means of control messages called Association CONFIguration (ASCONF). The SCTP can inform the CN, that a new IP address is available. Providing a tool to add backup IP paths used in the case of failure of the primary path was the early design aim of the ASCONF. Recently, this feature has been exploited to switch the communication among different paths as in the case of mobile communications [12]. For Web-like applications, the MN directly inform the Web server of the new acquired IP address. There is no need of a location server for the MN. The u function is directly executed by the MN. In this case, the MN uses the u function with $x\text{-loc} = CN_{IP}$, that is the IP_{new} of the MN is directly updated at the CN. The g function depends on the application: If the MN

²We use the words *caller* and originating entity as synonymous, although the former word is suitable for multimedia communications.

Table 1
Comparison of mobility functions that each protocol can execute

Application	Protocols	CN \mapsto CN	CN \mapsto MN	MN \mapsto CN	MN \mapsto MN
Web FTP	SCTP	(u)	u	u	u
	SIP	(f, g, u)	(f, g, u)	(f, g, u)	f, g, u , needs statefull server
	MIP	(f, g, u)	(f, g, u)	(f, g, u)	f, g, u , needs changes to MIP-LR
VoIP	SCTP	(u)	u	u	u
	SIP	f, g, u	f, g, u	f, g, u	f, g, u , needs statefull server
	MIP	f, g, u	f, g, u	f, g, u	f, g, u , needs changes to MIP-LR

is the originating entity towards a not moving one (e.g. an FTP download), g is not needed, because as said f furnishes all the necessary information to localise the Web or FTP server; on the other hand, if the MN is the *callee*, as in VoIP applications, a location server must be employed. In the following, we will assume SCTP as transport for any kind of communication, because it can be used also for real-time communication, by using its partial reliability variant [14].

As a matter of comparison, we summarise the mobility cases above discussed in Table 1, where the communication direction symbol, $x \mapsto y$, means that the x is the originating entity and requires data from y , that is the *callee*. In the table, for each protocol and communication direction, a (\cdot) means that the function is not strictly necessary for the mobility procedures.

In the following we give more details.

2.1. MIP+SIP

In this case, for VoIP applications, the network is redundant, because both SIP server and MIP's HA perform the same localisation function, f , as shown in Table 1. As the name suggests, VoIP applications are indeed applications, they inherently need a location server to perform function f . Thus, the use of MIP is redundant. This fact has been observed also in [8], where the authors proposed to use SIP registration whenever it is necessary, although it is not clear if the HA should be integrated inside the SIP servers or not. For non-VoIP applications, the function f , if required, is executed by standard mechanisms like DNS (second and third row in Table 1). Generally to locate the MN, a SIP server should be employed. Let us note that if RO is not present, MIP does also function g (i.e. the packets forwarding), increasing the overhead,³ the delays and packet losses, both during the hand-off and during the communication.

2.2. MIPv6+SIP

Again, the network is redundant. On the other hand, MIPv6 avoids the triangular routing problem by means of the direct Return Routability Procedure (RRP) to the CN. Decapsulation is not strictly needed because the CoA address is globally routable. Here, the latency budget depends on the Duplicate Address Detection (DAD) procedure which can take long time. DAD guarantees that the new generated IP address is unique. We suppose that the MN can acquire a new CoA by means of a DAD Cache server, which performs in advance the DAD procedure [4].

³Due to packets decapsulation and triangular forwarding.

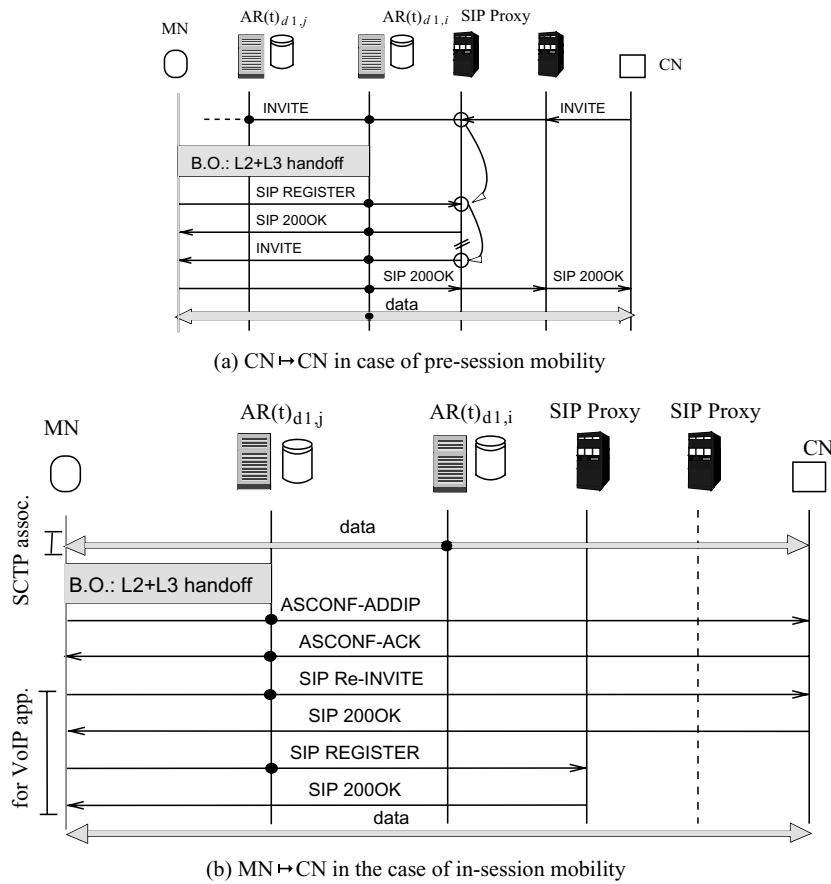


Fig. 1. Mobility in the case of single movement.

2.3. SCTP+IPv6+SIP

2.3.1. Single movement

In this architecture, the SIP server is aware of the most updated address of the MN. Whenever the MN enters a new subnetwork, an L2/L3 blackout (BO) takes place, as shown in Fig. 1, and the MN acquires a new CoA by means of IPv6 address auto-configuration. In Fig. 1-a, the CN initiates the communication (VoIP application) towards the MN by means of the SIP INVITE message. The CN performs the function f by querying its SIP server (or the SIP server of the MN if both peers belong to the same domain). Since the location stored in the MN's SIP server could be not yet updated, the SIP server should immediately send a pending INVITE whenever it receives a REGISTER message from the interested MN, as illustrated in the pseudo-code of Algorithm 1. In fact, usually the REGISTER message is sent to the SIP server when the SIP application is made aware of a change of the IP address.⁴ The alternative is waiting for timers expiration which will delay the call setup. This could be a serious problem if the MN hand-off rate is high. Algorithm 1 means that the SIP server should be statefull. In both cases, for pre-session mobility the SCTP does not play a crucial role. On the other hand, if the

⁴This requires dedicated support from the kernel sockets.

MN starts non-VoIP communication towards the CN, as in Fig. 1-b, the SCTP transport will manage the movements of the MN by means of ASCONF-ADDIP and ASCONF-ACK messages. No other components are needed.

In this architecture (SIP+SCTP+IPv6), redundant registrations are not present. The additional components are: the SIP servers, and the Access Router (AR) supporting IPv6 which is always present in (wireless) subnetwork. One could observe that a full SIP architecture could be enough. But for non-VoIP applications, the functions f, g could not be performed because the CN could not have neither a SIP-based domain nor a SIP-compliant application. Our opinion is that SIP should be used whenever the application requires it, while in all other cases the mobility management should be addressed to the transport layer. Another problem of the application based mobility management is that the connection must be re-opened after the Re-INVITE. In fact, current socket APIs do not allow to change the destination IP address of a running connection. For UDP transport, the additional delay to close and re-open the socket roughly depends on the operating system. In Fig. 1, we do not explicitly show where and when the SCTP association is opened. In fact, this is a problem that requires a more detailed explanation which we do not face here. However, we suppose that the association is opened after both peers have been updated with location information each other. This means that the association is opened after the 200OK message has been received. Accordingly, for VoIP applications, if special Real Time Protocol translators are not present in the SIP domain, the association must be re-opened (“break and re-make”). This could be deleterious for the quality of the communication. The use of SCTP can be favourable if one thinks that standard applications like FTP and Web downloads perform better than their counterparts based on TCP, as pointed out in [7]. Moreover, SCTP’s SACK mechanism reduces the number of timeouts caused by radio links impairments.

2.3.2. Simultaneous movements

In this case, both peers are MNs. We suppose that their locations are registered at the SIP server. For example, suppose that at time t_1 the MN enters a BO phase, and after the L2/L3 hand-off, the MN is able to forward and receive packets from $AR(t_1)_{d1,i}$, as in Fig. 2-a-b, that is the i -th AR visited at time t_1 and belonging to domain $d1$. At time t_2 , the MN in domain 2 moves towards the subnetwork of $AR(t_2)_{d2,i}$ with $i \neq j$, and at time t_3 the MN in domain 1 moves towards $AR(t_3)_{d1,j}$. In other words, the MNs can travel along different subnetworks, i, j , within the same domain, $d1, d2$. To handle these simultaneous movements of both peers, we have two possible solutions.

2.3.2.1. Full SIP solution

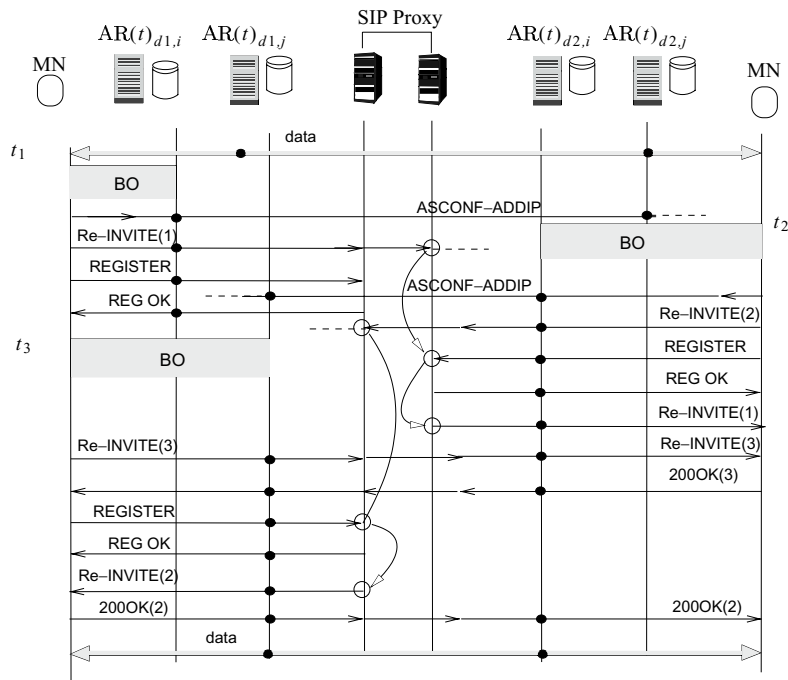
The first solution requires the use of a Re-INVITE message, as in Fig. 2-a, because the ASCONF-ADDIP is not able to perform neither the f function nor the u function because the ASCONF uses an invalid destination address, that is the IP address of the correspondent peer before its movement. In this case, the use of statefull SIP server is mandatory in order to decrease hand-off latency and avoid possible signalling problems, e.g. deadlock states of the hand-off procedures, as also pointed out in [17]. Again, the Algorithm 1 should be executed.

Algorithm 1.

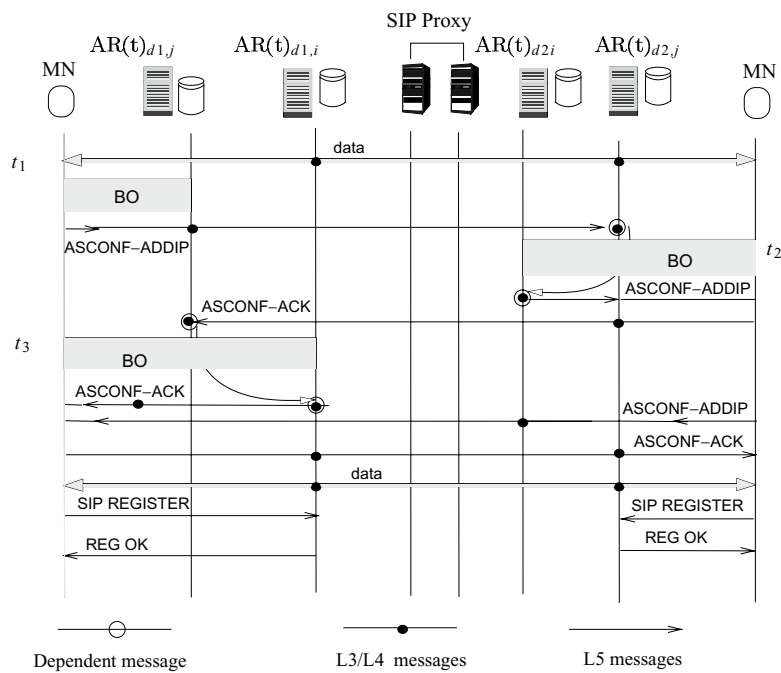
```

 $U_i$  =  $i$ -th User or SIP application
 $PM_i$  = list of pending messages for  $U_i$ 
 $m$  = message in  $PM_i$ 
for each REGISTER from  $U_i$ 
if  $PM_i \neq \emptyset$  then
send  $\forall m \in PM_i$ 

```



(a) MN \leftrightarrow MN with statefull SIP server



(b) MN \leftrightarrow MN with smart AR

Fig. 2. Mobility in the case of simultaneous movements.

2.3.2.2. SCTP solution

SCTP is unable to handle simultaneous movements, because ASCONF messages use destination IP addresses which are not valid at the hand-off time. This problem is common to any end-to-end solution like mobile SCTP. Here, we recognise another way to handle both simultaneous and single movement cases, by re-using network components always present. Although we do not address in detail this solution here, it is worth giving its basic concepts in order to compare it with other solutions. In Fig. 2-b, we show this solution, which does not require the Re-INVITE message, because of the use of SCTP. Moreover, the use of a modified AR avoids possible breaking of the communication. We suppose that in every domain the $AR(\cdot)_{dk,i}$, $k = 1, 2$, is smart, that is it can store for a certain period of time the new location of the MN just departed. For example, the AR can store an addresses cache for every MN, whose entries are the available addresses of an MN along with their state. Whenever the MN leaves from the subnetwork $AR(\cdot)_{dk,i}$, the cache for that MN will change the state of the old address, e.g. from a “Preferred” state to a “Deprecated” one,⁵ and a new “Preferred” address is being added into the cache, i.e. the MN’s address acquired in the subnetwork $AR(\cdot)_{dk,j}$. By this way, the $AR(\cdot)_{dk,i}$ can forward packets whose destination is “Deprecated” to the proper “Preferred” address, e.g. from $AR(\cdot)_{dk,i}$ to $AR(\cdot)_{dk,j}$. We do not enter in the details of the protocol, but only note that this mechanism could be coupled with the Cache server discussed previously.

3. Performance analysis

As seen, the performance of any micro-mobility protocol is affected by the hand-off management. The performance factors of the hand-off mechanism are the latency and the packet loss. The latency depends on the BO phase and the number of signalling messages which are to be sent in order to get the right IP address for both peers. Moreover, for reliable transmission, as in standard SCTP, the latency also depends on the delays of loss recovery. In fact, during the hand-off some packets could not arrive at the destination because of the BO phase; consequently, timeouts could take place at the sender side. Therefore, latency can be defined as the time interval between the last successful transmission time on the old path and the transmission time of the first new packet on the new path. In this section, we analyse the hand-off latency for the mobility schemes discussed above. However, we concentrate the study on SCTP only. We give a comparison of the standard with our proposed technique. Furthermore, as we anticipated in Section 1, we show that the probability of simultaneous movements is not negligible. This fact emphasises the need of using a localisation service, like that furnished by the SIP architecture.

3.1. Mobile SCTP

Firstly, we note that the use of ADDIP parameter in wireless communication is not as easy as in the case of wired communication. In fact, in the latter case the SCTP protocol receives the ASCONF-ADDIP message from the primary path (i.e. the old path in our scenario) and the transmission can advance without interruption. In wireless transmission, the ASCONF message cannot be sent on the primary path because this path is unavailable unless the wireless network card allows multiple communications with more than one AP. Accordingly, packet losses and timeouts can occur. Moreover, the transmission over an unknown path should begin with the slow start phase of the congestion control, even if there are

⁵This mechanism is similar to the addressing scheme of IPv6.

enough packets to trigger a Fast Recovery (FR) by sending all the remaining packets over the new path. The solution in [2] meets this constraint, while the authors in [1], without any analysis, suggest to use half of the congestion window on the previous path as the first value of congestion window on the new path. We use the slow start phase constraint.

Algorithm 2.

```

for each received ASCONF-ADDIP
# Pj = new j-th path
# ( )i = variables on the previous path
  if (set_primary) & (gap == null)  stop all timers;
  # Recovery phase in slow start
  if version1:  send(highest(TSNi)) on Pj;
  if version2: send(OutstandingTSNi) on Pj;

```

In the case of reliable transmission, the latency is made of two terms: the first one is the delay the sender waits until the arrival of the ASCONF message; the second one is the time of lost packets recovery phase, as shown in Fig. 3-a for standard SCTP. If no timeout occurs, we have re-transmission according with the congestion control as in Fig. 3-b. We now analyse the latency. Let's be T_{H_n} the latency in the case of timeouts. For the new path, the transmission delays of the MN->CN and CN->MN paths are Δ_{new}^f and Δ_{new}^r , respectively. The BO phase lasts T_{BO} seconds. The number of timeouts during the handover is a random variable, N ; similarly, the number of lost packets is L and the duration of the recovery phase is L_r . For standard SCTP, we note that if the ASCONF-ADDIP message arrives after a timer is started, the protocol sends the next packet after a time R until the expiration of the timer, as in Fig. 3-a. In this case, the mean values are:

$$E\{T_{H_n}\} = E\{T_{BO}\} + E\{R\} + E\{L\} + \Delta_{new}^f + \Delta_{new}^r, \quad (1)$$

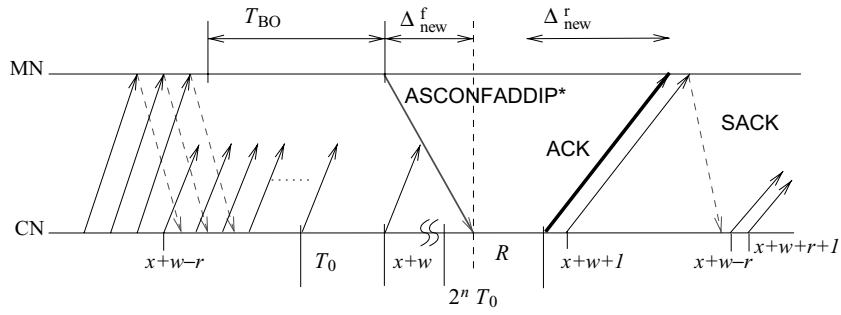
where we suppose $\Delta^{f,r}$ be deterministic values, and

$$E\{R\} = \sum_{n=1}^{n_{\max}} 2^{n-1} T_0 P\{N = n\} \quad (2)$$

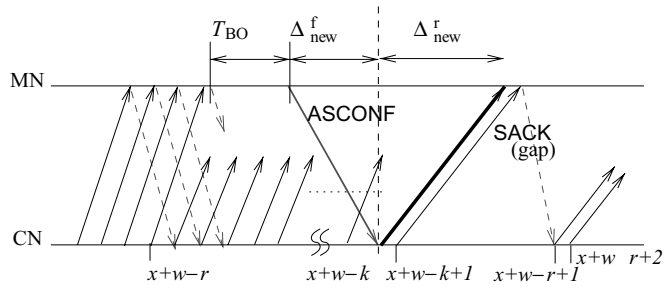
$$E\{L_r\} = \sum_{l=1}^{l_{\max}} RTT [\lceil \log_2 l \rceil + 1] P\{L = l\}, \quad (3)$$

where RTT is the round-trip time of the new path, T_0 is the base timeout value of SCTP. In general, values of the BO phase and the one-way delays are under 500 ms and $T_0 = 3$ s. The distribution of the T_{H_n} , R , and L depends on the residual time of a current transmission until the emptying of the transmission window. We analyse this dependence in Section 3.2. If one accounts also the delays of DAD procedures, the BO phase can be much longer and the probability of more than one timeout increases. Moreover, the logarithmic term in Eq. (3) is due to the slow start phase during which the number of sent packets is doubled every RTT .

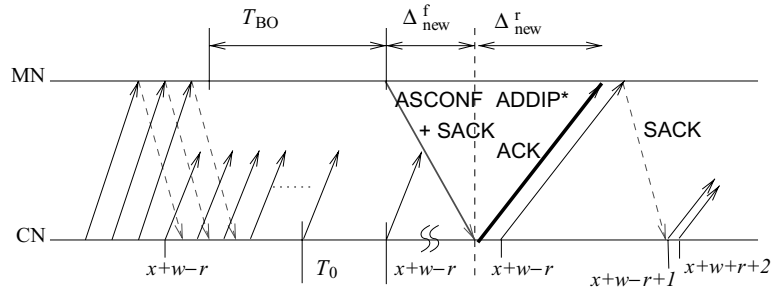
If we use the “version 1” of Algorithm 2, the mean of L_r is as in the sum of Eq. (3), which is the solution of [2]. In that work the packet with $TSN = x + w + 1$ is called probe packet, where w is the congestion window before the hand-off. However, in that work is not clear if the ASCONF-ACK is sent



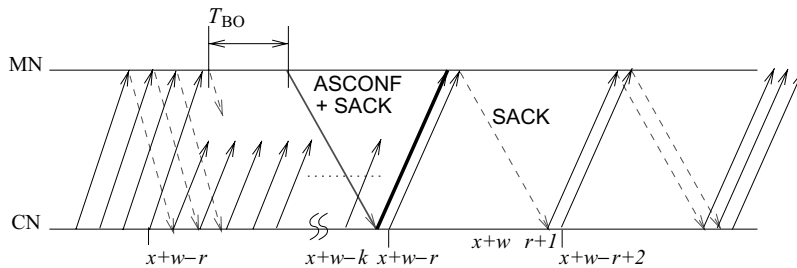
(a) Handoff with timeouts in SCTP standard



(b) Handoff without timeouts



(c) Optimization of latency in case of timeouts, "version2" of Alg.2



(d) "Version2" of Alg.2 without timeouts

Fig. 3. Loss recovery for different management of ASCONF-ADDIP.

or not, what is required by the draft document on ASCONF extensions to SCTP [15]. In our further Optimization, we use the “version2” of Algorithm 2, and then we have:

$$E\{L\} = \sum_1^{l_{\max}} RTT \lceil \lceil \log_2 l \rceil \rceil P\{L = l\},$$

that is, we can reduce the loss recovery time by one RTT . This is possible because the “version2” of the algorithm says that if the ASCONF-ADDIP intends to change the primary path immediately and if the SACK, bundled in or following the ASCONF, contains a null gap, then the old path is no longer available and the outstanding packets are surely lost. Then, the go-back- n procedure can be promptly activated. This situation is depicted in Fig. 3-d,e. We note that both in “version1” and “version2” of the Algorithm 2, $E\{R\} = 0$. Eventually, if no timeout takes place, we have $E\{R\} = 0$.

3.2. Numerical results

In order to compare our proposed Optimization to the base SCTP, we perform some basic simulations by means of MATLAB [16]. In particular, we simulate the hand-off phase of an MN in the middle of a file download. The file size is generated by sampling a Pareto random variable, with mean m and shape parameter β (see Appendix 4). For every sample, we compute in the order:

- The occurrence time of the hand-off, t_h ;
- the end of the current transmission⁶, t_f , and the residual time, $t_r = t_f - t_h$;
- the duration of the blackout phase⁷, $T_{BO} + \Delta_{new}^f$;
- the number of lost packets, L .

After this computation, we decide whether to perform a recovery phase in slow start or fast re-transmit mode. In the case of fast re-transmit, we simply assume that the recovery phase lasts a time T_r proportional to the transmission rate, C , and the number of lost packets during the hand-off. That is, we are assuming that, after the hand-off and the reception of the ASCONF message, there are enough packets to trigger, in one RTT , the fast re-transmit procedure. Accordingly, we have:

$$T_r = \Delta_{new}^r + \Delta_{new}^f + L * 8 * \frac{P_s}{C}, \quad (4)$$

where P_s is the packet size in bytes. In the case of timeouts, we compute the recovery phase delay for the base SCTP and our Optimization. With the above steps, we collect 10000 samples of the hand-off process and compute the Cumulative Distribution Function (CDF) of T_{H_n} , along with its mean, variance and median. The parameters of the simulation are summarised in Table 2. The results are shown in Fig. 4. As shown, with our technique, the mean value of T_{H_n} is halved. Let us also note that in the standard SCTP, the contribute of the R on the hand-off delay is remarkable: This is the reason of the up shift of the samples in the case of standard SCTP, Fig. 4-a,c.

⁶Let us note that if a hand-off happens, the end of “current transmission” means the time when all packets have been delivered to the link layer buffer. This is a simplification, because the right time the timer should be started from, depends on the current estimation of the timeout value.

⁷Here, we consider deterministic values only.

Table 2
Simulation parameters

C	P_s	T_{BO}	T_0	$\Delta_{new}^{f,r}$	m , in packets	β
1 Mbps	1 kbytes	500 ms	500 ms	500 ms	1.4	1.2

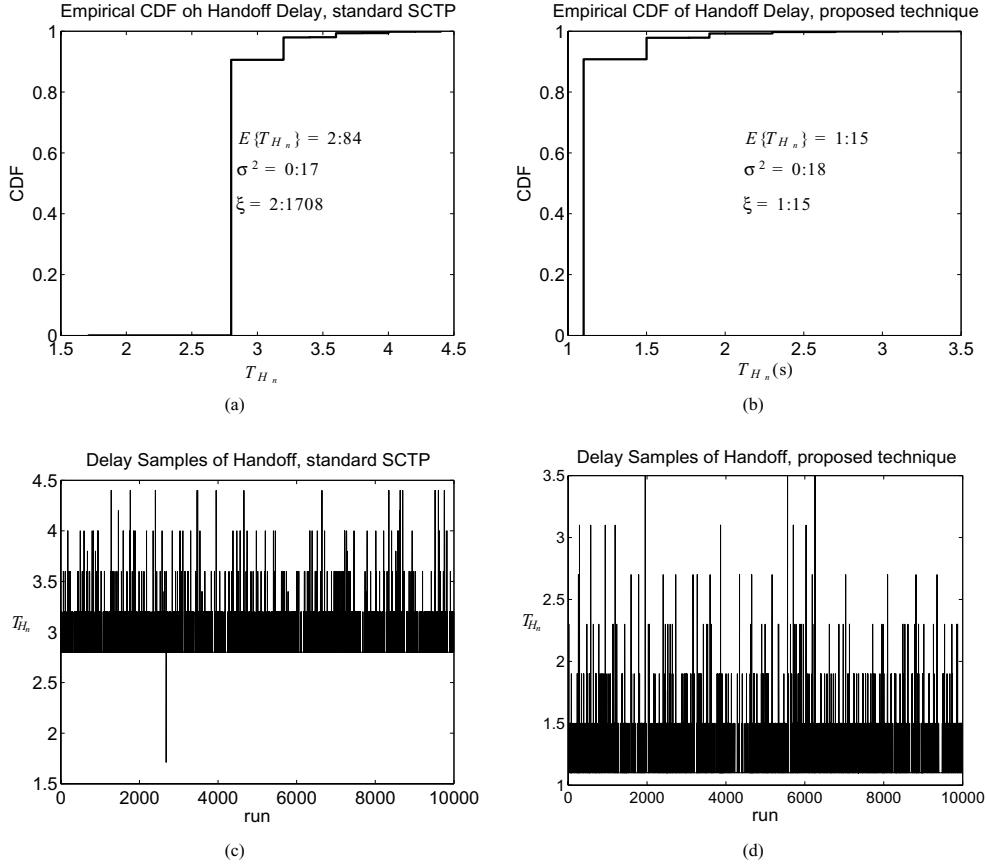


Fig. 4. Numerical results.

3.3. Simultaneous movements

In the case of simultaneous movements, the BO phase perceived by each peer can be longer. In the following, we suppose to use Algorithm 2. For simplicity, we focus on the case of Fig. 2-d and we call MN_1 and MN_2 the MN in domain 1 and 2, respectively. For simplicity, we omit the subscript *new* by supposing that the reverse and the forward path delays are equal for the old and the new path, respectively. We assume the communication delays between the ARs are negligible and for both MNs the delays of BO phase are identically distributed. We name *Sim* the event of a deleterious simultaneous movement, i.e. the event in which both MNs use an erroneous IP address for the destination of packets. The probability of this event is $P(Sim) = p_S$. Now, we can write Eq. (1) as follows.

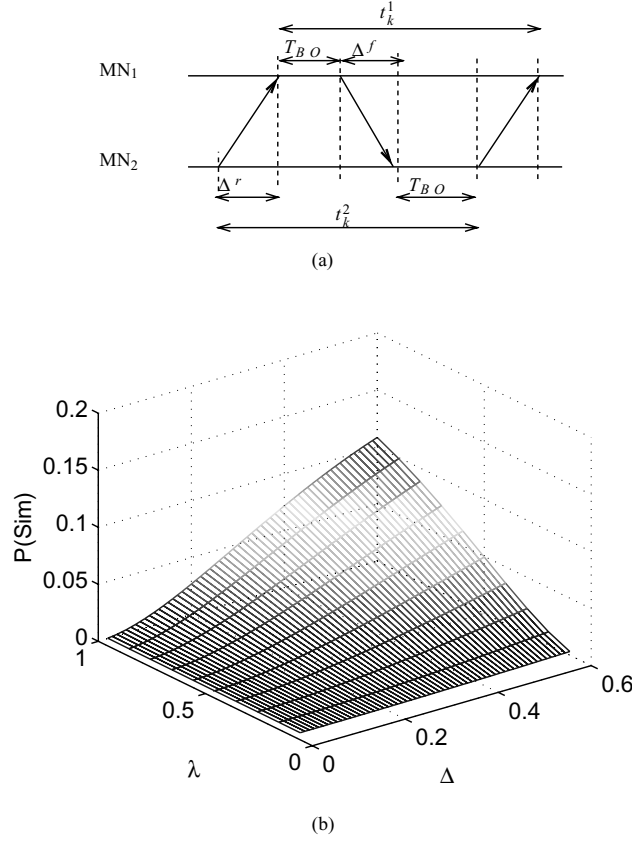


Fig. 5. In a) critical simultaneous movements. In b) simultaneous movements probability.

$$E\{T_{H_n}\} = p_S E\{T_{H_n}|Sim\} + (1 - p_S) E\{T_{H_n}|\overline{Sim}\}, \quad (5)$$

where $E\{T_{H_n}|\overline{Sim}\}$ is as in Eq. (1), while

$$E\{T_{H_n}|Sim\} = E\{L|Sim\} + \Delta^f + \Delta^r + 2E\{T_{BO}\} \quad (6)$$

and

$$E\{T_{H_n}\} = p_S E\{L|Sim\} + (1 - p_S) E\{L|\overline{Sim}\} + \Delta^f + \Delta^r + (1 + p_S) E\{T_{BO}\}. \quad (7)$$

The probability of Sim depends on the interval the MNs spend to change the IP address and the rate of the hand-off, i.e. $\lambda_1 = \frac{1}{E\{t_k^1\}}$ and $\lambda_2 = \frac{1}{E\{t_k^2\}}$. From Fig. 5-a, it is easy to recognise that the critical interval during which a simultaneous movement can happen is $\Delta^r + \Delta^f$. In fact, the ADDIP of MN₂ arrives at MN₁ in after a time of Δ^r and similarly for the ADDIP of MN₁. Thus, the probability of Sim is the probability of an hand-off arrival in these intervals. If we suppose that the hand-off inter-arrival times are exponentially distributed with means λ_1 and λ_2 for MN₁ and MN₂, respectively, we have:

$$p_S = e^{-\lambda_1 \Delta^f} e^{-\lambda_2 \Delta^r} \lambda_1 \lambda_2 \Delta^f \Delta^r.$$

In Fig. 5-b, we plot an example of this probability, for $\Delta^{f,r} = \Delta$ and $\lambda = \lambda_{1,2}$. As shown, the probability of simultaneous movements is not negligible as the hand-off rate and the path delays increase [17]. In

general, the hand-off rate depends on the size of the radio cell and the velocity of the MN and can be *high* especially in ubiquitous applications like inter-vehicle communications. We observe that p_S is also the probability of communication drop if the smart ARs are not employed. In the simultaneous movements case, $E\{L|Sim\}$ is more complicated, because the number of lost packets depends also on the direction of the communication. For example, if MN_1 , which is transmitting data towards MN_2 , performs the first movement, the data are not lost but buffered at the peer; after its hand-off is completed, the buffered data will be transmitted on the new path. From this point, losses can happen because of the second movement, i.e. that of MN_2 .

3.4. SIP

For single movements, as in Fig. 1-b, the latency is as in Eq. (1), because the SIP Re-INVITE and SIP 200OK messages are the equivalent of ASCONF-ADDIP and its ACK, respectively. The slight difference would be that the socket needs not to be re-opened. For simultaneous movements, the latency can be longer. In fact, as shown in Fig. 2-a, the MN_2 must wait the arrival of the SIP Re-INVITE message in order to “re-open” the connection with the new IP of MN_1 . This message is sent by SIP server of MN_2 , after the reception of the REGISTER message: if the SIP proxy is far, the latency will be higher than that of the SCTP case. In other words, the affected parameters are Δ^r and Δ^f . Accordingly, the latency would be:

$$E\{T_{H_n}\}_{SIP} \geq E\{T_{H_n}\}_{SCTP},$$

with obvious meaning of the subscript.

4. Conclusions

In this paper, we discussed pro and cons of current protocols for the management of micro-mobility in IP wireless networks. First, we gave a classification of the mobility problems based on mobility functions. We recognised that some combinations, like MIP+SIP, introduce redundancy in the network. Moreover, with a kind of criticism, we saw that if the standard applications, like Web and FTP, are not supported by SIP based domains, the micro-mobility can not be realised with SIP. On the other hand, transport layer solutions to the micro-mobility problem are possible whenever the protocol permits the dynamic change of the destination IP address. The first attempt to end-to-end mobility was the TCP-Migrate protocol [13], which used the updating of DNS to control the location of the MN. Here, we used SCTP with the ASCONF messages support. This solution does not require SIP domains. However, since the ASCONF has not been designed for micro-mobility minimum changes should be taken into account. In particular, in Section 2, we analysed a possible Optimization of the hand-off latency when ASCONF is used. We shown as first optimisation a smarter handling of timers: if the ASCONF contains a “set-primary” parameter, they should be stopped, otherwise there is a mean term of T_0 in the latency, at least when the congestion window is less than 2 packets. The T_0 value is the base timeout value of SCTP, usually set to 3s. The second optimisation is on the loss recovery time. We proposed to re-start go-back- n procedures of SCTP whenever an ASCONF (with “set-primary”) is followed by an SACK packet with null gap report. This two messages are a notification of hand-off and the probe packet is not needed. We moved forward in the analysis by considering the case of simultaneous mobility. The analysis in Section 3 pointed out the following results:

- The optimisation of ASCONF is much more important in this case, because T_{B0} can be much longer and then $E\{R\}$ could be greater than T_0 .
- The use of SCTP and any other transport solution requires a location service; another solution could be the smartAR, as described in Section 2.3.2.
- The use of SIP requires statefull SIP server and the Algorithm 1.
- The more the SIP servers are *far* from the subnetwork of the MN, the more the use of SIP increases the latency of the hand-off.

As a matter for further investigation, we note that subnetworks equipped with the smart AR could solve both problems of SCTP, for the updating function, and SIP, for the decrease of the hand-off latency.

Appendix

In the simulations, we used a Pareto distribution in order to generate the length of a file. A Pareto random variable (r.v.) X has the following CDF:

$$F_X(x) = P[X \leq x] = 1 - \left(\frac{x}{\alpha}\right)^{-\beta}, \quad (8)$$

where $\alpha > 0, \beta > 0$. Its mean is:

$$m \triangleq E\{X\} = \frac{\beta\alpha}{\beta - 1}. \quad (9)$$

This Pareto distribution is heavy-tailed, that is the probability $P[X > x] \neq 0$, for $x \rightarrow \infty$. Moreover, this distribution is for continuous variables. We should use the Zipf distribution, because the file length is a discrete r.v. However, we simply generated a Pareto r.v. and computed the length of the file by ceiling the decimal part. In the simulations, we used the mean and shape parameter. Then, we had to generate a Pareto distributed r.v. with CDF:

$$F_X(x) = P[X \leq x] = 1 - \left(\frac{x\beta}{m(\beta - 1)}\right)^{-\beta}. \quad (10)$$

To generate such a variable, we used the integral transform method [10]. That is, we generate an uniform r.v. in $[0, 1]$, U ; then, we used $F_X^{-1}(U)$ to compute X :

$$X = \frac{m(\beta - 1)}{\beta}(1 - x)^{-\beta}. \quad (11)$$

References

- [1] I. Aydin, W. Seok and C.-C. Shen, *Cellular sctp: a transport layer approach to internet mobility*, in Proceedings of IEEE ICCCN'03, October 2003, 285–290.
- [2] M. Chang, H. Lee, Y. Hong, M. Lee and J. Park, *An enhancement of transport layer approach to mobility support*, in Proceeding of ICOIN'05, LCNS, Springer-Verlag, 2005, 864–873.
- [3] C. Guo, Z. Guo, Q. Zhang and W. Zhu, *A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks*, *IEEE Journal on Selected Areas in Communications* **22**(5) (2004), 834–848.
- [4] S.-H. Hwang, Y.H. Han, C.-S. Hwang and S.-G. Min, *An address configuration and confirmation scheme for seamless mobility support in ipv6 network*, in Proceedings of Wired/Wireless Internet Communications Conference, volume 2957 of LCNS, Springer-Verlag, 2004, 74–86.

- [5] Ietf. Sip working group. 2005.
- [6] D. Johnson, C. Perkins and J. Arkko, *Mobility Support in IPv6*, IETF RFC 3775, 2004.
- [7] S. Landha and P. Amer, *Improving multiple file transfers using sctp multistreaming*, in IEEE Int'l Performance, Computing and Communication Conference (IPCCC), volume 4, 2004, 513–522.
- [8] 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 Proceedings of IEEE Vehicular Technology Conference, volume 3, October 2003, 2058–2062.
- [9] A. Mishra, M. Shin and W. Arbaugh, An empirical analysis of the iee802.11 mac layer handoff process, *ACM Computer Communication Review* **33**(2) (2003), 93–102.
- [10] A.M. Mood, F.A. Graybill and D.C. Boes, *Introduction to the Theory of Statistics*, McGraw-Hill, 1974.
- [11] N. Nakajima, A. Dutta, S. Das and H. Schulzrinne, *Handoff delay analysis and measurement for sip based mobility in ipv6*, in Proceedings of ICC'2003, volume 26, 2003, 2058–2062.
- [12] M. Riegel and M. Tuexen, *Mobile sctp*, IETF – draft, 2003.
- [13] A.C. Snoeren and H. Balakrishnan, *An end-to-end approach to host mobility*, in Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom), 2000, 155–166.
- [14] R. Stewart et al., *Stream Control Transmission Protocol (SCTP) Partial Reliability Extension*, IETF – RF3758, May 2004.
- [15] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen and I. Rytina, *Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration*, IETF – draft, December 2004.
- [16] The Mathworks, 2006. www.mathworks.com.
- [17] K.D. Wong, A. Dutta, K. Young and H. Shulzrinne, *Managing simultaneous mobility of ip hosts*, in Proceedings of MILCOM'03, volume 22, 2003, 785–790.

Giuseppe De Marco received the degree (“laurea”) in Electrical Engineering in 1999 from University of Bologna, Italy. He received the PhD degree from University of Salerno, Italy, in 2004. Currently, he is a JSPS Post Doctor Fellow Researcher with the Department of Communication Engineering, at the Fukuoka Institute of Technology, Japan. His main scientific interests are protocol design, performance analysis, wireless networks and heuristic techniques applied for the optimization of packet networks. He is a member of IEEE.

Leonard Barolli received BE and PhD degrees from Tirana University and Yamagata University in 1989 and 1997, respectively. From April 1997 to March 1999, he was a JSPS Post Doctor Fellow Researcher at Department of Electrical and Information Engineering, Yamagata University. From April 1999 to March 2002, he worked as a Research Associate at the Department of Public Policy and Social Studies, Yamagata University. From April 2002 to March 2003, he was an Assistant Professor at Department of Computer Science, Saitama Institute of Technology (SIT). From April 2003 to March 2005, he was an Associate Professor and presently is a Full Professor, at Department of Information and Communication Engineering, Fukuoka Institute of Technology (FIT). Dr. Barolli has published more than 150 papers in referred Journals and International Conference proceedings. He was an Editor of the IPSJ Journal and has served as a Guest Editor for many International Journals. Dr. Barolli has been a PC Member of many International Conferences and was the PC Chair of IEEE AINA-2004 and IEEE ICPADS-2005. He was General Co-Chair of IEEE AINA-2006 and is Workshops Co-Chair of MoMM-2006 and IEEE AINA-2007. His research interests include network traffic control, fuzzy control, genetic algorithms, agent-based systems, ad-hoc networks and sensor networks. He is a member of SOFT, IPSJ, and IEEE.

Salvatore Loreto received the laurea degree (5 years) in software engineering from University of Naples, Italy, in 1999. He is pursuing the Ph.D. degree in Computer Science at University Federico II, Naples. Currently, he is with Ericsson Research, Finland. His main scientific interests are signaling protocols, P2P networks and network software



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

