A Performance Study of Context Transfer Protocol for QoS Support

Novella Bartolini[1], Paolo Campegiani[2],
Emiliano Casalicchio[2], and Salvatore Tucci[2]

[1] Universitá di Roma "La Sapienza", Via Salaria 113 - 00198 Roma, Italy
novella@dsi.uniroma1.it
[2] Universitá di Roma "Tor Vergata", Via del Politecnico, 1 - 00133 Roma, Italy
{campegiani,casalicchio,tucci}@ing.uniroma2.it

Abstract. In nowadays wireless networks, mobile users frequently access context dependent Internet services. During handover procedures, the management of context related information introduces additional overheads to transfer context-aware service sessions. The overhead due to context transfer procedures may affect the quality of service perceived by mobile users making more difficult to realize seamless handover procedures. Context Transfer Protocol can improve the QoS perceived by mobile nodes that access context dependent services. In this paper we extend motivations for context transfer, and we introduce three different scenarios for Context Transfer Protocol. We propose a performance model to compare these scenarios when context transfer protocol run on top of IPv6 with fast handover mechanisms.

1 Introduction

Nowadays internet services are often session oriented, delay bounded (or real-time) and context sensitive. Just to mention some, VoIP, multimedia streaming, on-line games, on-line transactions and many Content Delivery Networks related services are often session oriented, delay bounded and context sensitive.

In wired networks, the use of broadband technologies has a significant impact on the user perceived Quality of Service (QoS) making it possible to fulfill Service Level Agreements (SLA). On the contrary, in wireless networks the introduction of broadband wireless connectivity is not sufficient to guarantee the fulfillment of QoS requirements mostly due to users movement across network coverage areas managed by different Access Routers (AR). Handover requests may be issued during critical service phases for which the avoidance of service disruption is mandatory, and the connection must be seamlessly handed off from a point of access to another.

The fast handover mechanism, introduced to reduce the packet losses during handovers, needs to be enhanced with proper mechanisms to preserve the session continuity. In context-aware services, handover is not only a matter of keeping a connection alive during users movements, but also of transferring the necessary information to avoid the re-establishment of a service session every time the user

C. Aykanat et al. (Eds.): ISCIS 2004, LNCS 3280, pp. 594–603, 2004.

reaches a new point of access. The re-establishment of a service session causes the repetition of the service initiation message flow from scratch and, most of all, the unavailability of the necessary information to keep the service alive without the need of a restart. Thence session continuity and context transfer during handover procedures are very critical for delay sensitive and context dependent application. We extend the general motivation for context transfer identified by the IETF SeaMoby working group [5]. Exchanged informations could relate to:

- authentication, authorization, and accounting information [5] needed to permit the re-authentication of the mobile host and the mobile host's authorization to access the network service from a new subnet;
- header compression [5] information that is necessary to avoid the repetition of messages between the last hop router and the mobile host with full or partially compressed headers before full compression is available;
- network QoS information to avoid the re-negotiation and re-establishment of QoS agreements between the mobile node and routers;
- application level QoS parameters, e.g. maximum end-to-end perceived latency, level of image resolution (e.g. high-level resolution for laptop and low-level resolution for enlarged mobile phone/palmtop), maximum/minimum bit-rate for streaming sessions, security specification (e.g. which suite of encryption algorithms is allowed/used) service authentication (e.g. certificate, list of certification authorities, list of trusted servers);
- session state information, e.g. the number of items in the basket or the phase that most likely will be entered next, for an e-commerce session or the next chunk of data needed in a streaming session, the next game phase for an on-line game session, the mailbox state or the file system information of an e-storage account.

In all these scenarios, if procedures were conducted without transferring any context related information, descriptive parameters should be re-defined from scratch whenever the mobile host reaches a new access point. The re-negotiation of these parameters is too complex and may require longer time than the one that is needed to perform the handover. The best solution is to transfer context from the access router of the region from which the mobile node is coming (pAR) to the access router of the area targeted by the mobile node (nAR).

In section 2, we show the interaction between Context Transfer Protocol (CTP) [7] and Mobile IPv6 protocol, with fast handover mechanisms to reduce packet losses. In section 3 we describe the CTP message flow in some relevant cases. Section 4 shows a performance model of CTP using different metrics. Section 5 concludes the paper.

2 Mobility Management Mechanisms

To evaluate the impact of CTP on performance, its interaction with the underlying mobility management protocol must be considered and evaluated. Although an efficient and transparent mobility management mechanism affects every level

of the TCP/IP protocol stack, we consider only handovers that need to be managed at the network level (e.g. not analyzing handover occurring only at Data Link level) focusing on some relevant aspects of IPv6 and its fast handover mechanism.

Mobile IPv6 [4] defines the protocol operations and messages to achieve intra and inter-domain mobility within IPv6.

Auto-configuration [8][9] is an essential part of IPv6, and it's also used by a mobile node to obtain a new Care Of Address (nCOA) when it handovers to a new AR: the mobile node sends a Router Solicitation message (RtSol), and the AR responds with a Router Advertisement message (RtAdv) which contains the information needed by the mobile node to construct its nCOA as a global unicast address [2][3]. RtAdv messages are also broadcasted periodically. When the mobile node obtains its nCOA, it performs the return routability procedure, then it sends a Binding Update (BU) message to inform the CN of its nCOA. Only after these steps are concluded the CN and the mobile node can communicate directly, without routing triangularly via the Home Agent.

The time needed to complete this procedure is called handoff latency, and it is worth trying to reduce it as much as possible. This goal is pursued by the fast handover extension for IPv6 [6]. If this mechanism is put in place, the current AR not only broadcast its advertisement but also relays advertisements from confining ARs, by a Proxy Router Advertisement message (PrRtAdv), periodically broadcasted or sent out as an answer for a Proxy Router Solicitation message (PrRtSol), issued by a MN when it detects, by some layer two indicator, that a handoff is likely to occur.

Fast handover optimization also allows the MN to communicate its nCOA to the current AR (via a Fast Binding Update (FBU) message), so a layer 3 tunnel between the current AR (pAR, as the current AR is about to be the previous AR) and the new AR (nAR) could be established. This bidirectional tunnel is used to route packets from the nAR to the pAR.

It is worth pointing out that the whole fast handover mechanism can be applied only if the wireless interface is provided with an indicator of link layer level events such as discovering of a new AR or degradation of signal quality to the current AR.

3 The Context Transfer Protocol

When the mobile node moves to a new AR all context related data must be transferred from the previous AR and not obtained by an additional message exchange between the new AR and the mobile node, in order to avoid unnecessary bursts of data packets as the node gets connected to the new AR and to minimize the number of application data packets which cannot be processed properly due the lack of context-oriented information.

Context Transfer Protocol is composed of few messages: the Context Transfer Activate Request message (**CTAR**); the Context Transfer Request message

(**CTR**); the Context Transfer Data message (**CTD**); the Context Transfer Activate Acknowledge message (**CTAA**); the Context Transfer Data Reply message (**CTDR**); the Context Transfer Cancel message (**CTC**). For a complete description see [7].

The Context Transfer Protocol could be initiated by one of the ARs or by the mobile node, as a trigger (a Context Transfer Trigger) arises.

The pAR may initiate the CTP if it someway detects that the MN is about to handoff to another AR: in such a case it predictively sends the CTD message to the nAR.

The same could be done by the nAR when it detects that a mobile node is about to get connected to it: the nAR sends a CTR message to the pAR before the mobile node sends the CTAR message, so the CTD message (in reply to the CTR message) is received by the nAR before the time it would have been received if the nAR had waited for the CTAR message from the mobile node.

These first two scenarios are predictive, that is the context data transfer is initiated more or less before the actual handoff, and handoff latency is reduced.

The context transfer procedure can also be performed reactively: when the mobile node starts the handover at the data link layer, a CT Trigger arises so that the mobile node sends the CTAR message to the nAR, which in turn issues a CTR message to the pAR and receives from it the CTD message. This is a worst case scenario, showing the longest time to transfer the context.

3.1 Interactions between Fast Handover and Context Transfer Protocol

The context transfer is always triggered by means of a Context Transfer Trigger. The current version of the draft [7] doesn't define exactly what a CT Trigger is, although it seems to envision that the CT Trigger is a level two (data link) trigger. We believe that the CT Trigger could be better defined as a network level trigger. By doing so, we have a trigger which could be managed by the mobile node operating system, without requiring a hook provided by the wireless interface firmware.

The main idea is to use the Fast Handover messages as CT Trigger. As an example, if a pAR sends a PrRtAdv message to a nAR, it should also send a CTD to the nAR it is proxying advertisements for, because after the reception of a PrRtAdv message (or RtAdv) the MN is capable of performing an actual handover.

We can have these different scenarios:

Dummy Context Transfer Protocol. This is the completely reactive case when the fast handover mechanism doesn't take place, so the context transfer is initiated after the handoff of the mobile node from pAR to nAR: the nAR sends a RtAdv message to the mobile node which constructs its nCOA and sends a CTAR message to the nAR, which in turn sends a CTR message to the pAR. Figure 1(a) depicts this scenario, where no tunnelling is performed.

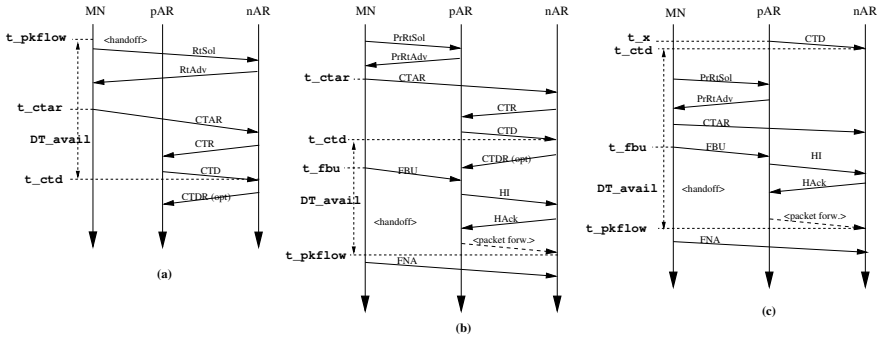

Fig. 1. Context Transfer Protocol scenarios: Dummy (a), Mobile Initiated (b) and Access Router initiated (c). The Definition of ΔT_{avail} in the three scenarios will be discussed in the performance analysis section

Mobile Node Initiated Context Transfer Protocol. The mobile node receives a PrRtAdv message from the pAR, and sends a CTAR to the nAR because it realizes that a handoff to the nAR is about to begin. It's worth noting that the mobile node could receive more than one PrRtAdv Message on behalf of different nARs, because the pAR could advertise (and usually do advertise) for all the confining nARs, and the mobile node could send the CTAR to one or more of advertised nAR, without knowing in advance which one it will handoff to (or *if* an handoff will take place): as the mobile node is still connected to the pAR, the pAR will receive all the CTAR messages and route them to the different ARs; if a targeted ARs respects the Context Transfer Protocol it will, after the reception of the CTAR, send a CTR to the current AR. Figure 1(b) shows the most favorable message flows for this scenario, when the actual handoff takes place after the context data have been transferred. The mobile node initiated case is designed to allow the new access router to use the context information to decide whether to manage or deny service to the new mobile node.

Access Router Initiated Context Transfer Protocol. The most predictive option is when the pAR (when it still is the current Access Router) sends a CTD describing a mobile node's context to one or more of its confining ARs. This can be done periodically or as a consequence of a CT Trigger. The receiving ARs cache this context, to be able to use it immediately after an handoff takes place. The context data are considered valid for a short period of time (possibly depending on the context type), after which they are removed; this soft-state approach is envisioned both for scalability and because the context data could (although slowly) change. Frequency of the CTD messages and cache duration must be defined accordingly to handoff frequency, available bandwidth for inter-AR communication and context data semantics.

Figure 1(c) shows the flow of messages when the pAR sends a CTD before the mobile node sends a PrRtSol message. Alternatively the pAR can trigger a PrRtSol message and send the CTD to the candidate nARs.

4 Performance Analysis of CTP

We introduce a performance model to evaluate the cost of CTP in terms of: consumed bandwidth and number of packets that have been lost or erroneously processed according to the default method, without considering the necessary context information.

At least three entities are involved in CTP: the mobile node, the previous access router and one or more new access routers. Thus we distinguish among the bandwidth consumed by the mobile node, $B^{\mathtt{MN}}$; the bandwidth consumed by the previous access router $B^{\mathtt{pAR}}$; and the bandwidth consumed by the new access router $B^{\mathtt{nAR}}$.

When a mobile node handovers to a new mobile access router, $N_{\mathtt{lost}}$ packets could be lost, and $N_{\mathtt{default}}$ packets could be erroneously served by default, without considering context related information. If an access router receives a packet before being able to consider the context related information, it processes the packet according to the default procedure, until the necessary information becomes available. When the AR receives context information and re-establishes the proper QoS level, packets will be properly prioritized.

4.1 Bandwidth Consumption Analysis

The Context Transfer Protocol works on an UDP-based transport layer. Our model is based on the assumption that CTP messages must fit the Maximum Segment Size (MSS) of a data link frame (and obviously must be contained in one UDP/IP packet), to reduce the packet fragmentation and reassembly overhead. For synchronization messages it is easy to fit the MSS, nevertheless context data could need a proper encoding and/or compression.

Each CTP message travels over an UDP segment, therefore the total overhead that is needed to send a CTP message is $O = O_{\mathtt{udp}} + O_{\mathtt{ip}} + O_{\mathtt{frame}}$, where $O_{\mathtt{udp}} = 8$ bytes, $O_{\mathtt{ip}} = 20$ bytes and $O_{\mathtt{frame}} = 18$ bytes (for ethernet frames).

In our analysis we give a formulation of upper bounds on the total amount of bandwidth consumed by each participant to perform the context transfer procedure. We use the following notation: $B^{\mathtt{participant}}_{\mathtt{scenario}}$ is the upper bound on the bandwidth consumed by $\mathtt{participant}$, where $\mathtt{participant} \in \{\mathtt{MN}, \mathtt{pAR}, \mathtt{nAR}\}$ and the triggering mechanism is $\mathtt{scenario} \in \{\mathtt{dummy}, \mathtt{MN}_{\mathtt{init}}, \mathtt{AR}_{\mathtt{init}}\}$.

In the following expressions S is the maximum size of the messages that are exchanged to perform the context transfer in the different scenarios. $s_{\mathtt{ctd}}$ is the size of the message containing context data and k is the number of new candidate access routers.

In the worst case, the pAR will complete the context transfer with all k candidates nARs. In a well designed architecture the nAR or pAR should abort

the context transfer when it is sufficiently clear that the mobile node will not enter the service area of the nAR.

We now formulate $B^{\texttt{participant}}_{\texttt{scenario}}$ for the different entities and different scenarios.

$$B^{\texttt{MN}}_{\texttt{dummy}} = 3(S+O), \tag{1}$$

$$B^{\texttt{pAR}}_{\texttt{dummy}} = [2(S+O) + (s_{\texttt{ctd}} + O)], \tag{2}$$

$$B^{\texttt{nAR}}_{\texttt{dummy}} = 3(S+O) + [2(S+O) + (s_{\texttt{ctd}} + O)], \tag{3}$$

$$B^{\texttt{MN}}_{\texttt{MNinit}} = (4+k)(S+O), \tag{4}$$

$$B^{\texttt{pAR}}_{\texttt{MNinit}} = 3(S+O) + \{k[2(S+O) + (s_{\texttt{ctd}} + O)] + 2(S+O)\}, \tag{5}$$

$$B^{\texttt{nAR}}_{\texttt{MNinit}} = 2(S+O) + [4(S+O) + (s_{\texttt{ctd}} + O)], \tag{6}$$

$$B^{\texttt{MN}}_{\texttt{ARinit}} = (4+k)(S+O), \tag{7}$$

$$B^{\texttt{pAR}}_{\texttt{ARinit}} = 3(S+O) + [k(s_{\texttt{ctd}} + O) + 2(S+O)], \tag{8}$$

$$B^{\texttt{nAR}}_{\texttt{ARinit}} = 2(S+O) + [2(S+O) + (s_{\texttt{ctd}} + O)]. \tag{9}$$

The first observation is that the bandwidth consumed at the MN (equations 1, 4 and 7) is directly proportional to the size of synchronization messages S in all scenarios, and also proportional to the number of k candidate nARs, in the mobile node initiated and access router initiated scenarios. In mobile initiate and access router initiated scenario it's important to operate a correct prediction of feasible next access router thus to reduce the bandwidth consumed at the MN that typically have no much bandwidth available.

The second characteristics of $B^{\texttt{nAR}}_{\texttt{scenario}}$ is that the bandwidth consumed at the nAR (equations 3, 6 and 9) is directly proportional to the size of context data $s_{\texttt{ctd}}$. The first term of equations 3, 6 and 9 gives a measure of the bandwidth consumed on the nAR-MN communication channel on the contrary, the second term, measures the bandwidth consumed on the pAR-nAR communication channels.

The third observation is that the bandwidth consumed by the pAR, in the last two scenarios, is a function of the number k of candidate nARs and of the size of context data $s_{\texttt{ctd}}$. The first terms of equations 5 and 8 give a measure of the bandwidth consumed on the pAR-MN communication channel, while equation 2 and the second terms of equations 5 and 8 measure the bandwidth consumed on the pAR-nARs communication channels, that is a function of $s_{\texttt{ctd}}$ in the dummy scenario and a function of $s_{\texttt{ctd}}$ and k in the mobile node initiated and access router initiated scenarios.

As a numerical example to give a quantitative idea of $B^{\texttt{participant}}_{\texttt{scenario}}$ we consider $S = 300$ bytes, $K = 4$ candidate access routers. This numerical example is shown in figures 2 and 3.

The mobile node initiate scenario is more bandwidth consuming than the access router initiated scenario because in the worst case the MN sends a CTAR message to each candidate nARs. In an analogous way, the pAR is the most stressed entity in terms of bandwidth because in the worst case the context will be broadcast to all the nARs that reply to the CTAR message or that are candidates. For the MN the dummy triggering mechanism consume less bandwidth that the other mechanisms, at the price of a degraded QoS. Figure 3 shows the trend of $B^{\texttt{participant}}_{\texttt{scenario}}$ when the number of candidate nARs increases (from 1 to 10) and the context data size has a fixed value 1020 bytes.

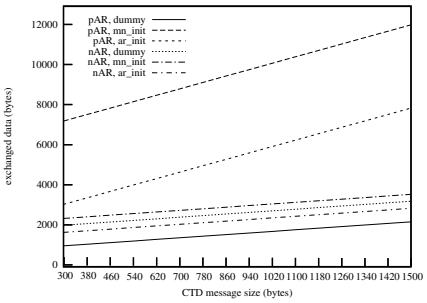

Fig. 2. Bandwidth consumed at the ARs by the CTP in function of the context data size

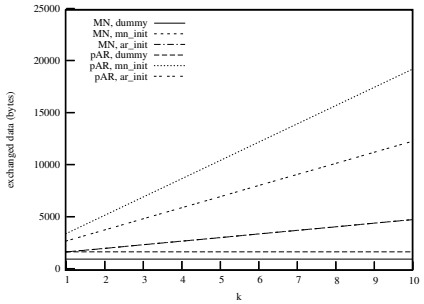

Fig. 3. Bandwidth consumed by the CTP in function of the number of candidate nARs and $s_{\texttt{ctd}} = 1020$ byte

4.2 Packet Loss and Bad Prioritization Analysis

Let r be the cumulative rate at which the mobile node and its related correspondent node inject packets into the network, and D the latency in the communication path between the mobile node and the correspondent node, through the pAR.

When a handoff occurs the MN registers itself in the new network and reestablishes the connection with the CN in $D_{\texttt{conn}}$ time units. In absence of a buffering mechanism between the pAR and the nAR, $N_{\texttt{lost}}$ packets are lost during handovers, where $N_{\texttt{lost}} = (t - t_{\texttt{hoff}}) \cdot r = D_{\texttt{conn}} \cdot r$, $t_{\texttt{hoff}}$ is the handover start time and t the instant of handover completion. On the contrary, if we use Fast Handover, packets are buffered by the pAR until a tunnel between the pAR and the nAR is established, therefore $N_{\texttt{lost}} = 0$.

In QoS sensitive applications, even a short sequence of packet loss could result in a SLA violation. For example a random packet loss can be tolerated in a low quality audio/video streaming session but it is prohibited in a secure transaction data flow.

In this paper we only focus on QoS sensitive application, where the condition $N_{\texttt{lost}} = 0$ is required, therefore the attention is restricted to the mobile initiated or access router initiated scenario. We refer to $t_{\texttt{ctd}}$ as to the instant in which the context is available to the nAR, and we refer to $t_{\texttt{pkflow}}$ as to the time the nAR starts processing packets directed from the CN to the MN. The elapsed time between the actual availability of the context data and the moment the first packets directed to the mobile node arrive to the nAR, can be expressed as $\Delta T_{\texttt{avail}} \leq (t_{\texttt{pkflow}} - t_{\texttt{ctd}})$.

As shown in figure 1 the context transfer begins at the instant $t_{\texttt{ctar}}$ in the mobile node initiated scenario and at time t_x in AR initiated scenario the nAR receives the context at time $t_{\texttt{ctd}}$, the handoff procedure starts at time $t_{\texttt{fbu}}$ and the nAR starts receiving packets addressed to the MN at time $t_{\texttt{pkflow}}$. When the context transfer procedure suffers from excessive delays and $\Delta T_{\texttt{avail}} < 0$, there is a period of time, that is $t_{\texttt{ctd}} - t_{\texttt{pkflow}}$, during which a certain number of packets belonging to an ongoing service, are erroneously treated by a default procedure, without considering context related information, thus causing a violation of the agreements on quality. The average number of packets erroneously treated by default is $N_{\texttt{default}} = -\Delta T_{\texttt{avail}} \cdot r = -(t_{\texttt{pkflow}} - t_{\texttt{ctd}}) \cdot r$. On the other side, if the handover procedure is completed on time, that is, if $\Delta T_{\texttt{avail}} \geq 0$, the SLA will be satisfied and $N_{\texttt{default}} = 0$.

We can conclude that a sufficient condition for the fulfillment of the SLA is $\Delta T_{\texttt{avail}} \geq 0$.

For lack of space we do not show the $\Delta T_{\texttt{avail}}$ model. A detailed dissertation is given in [1]. In the dummy scenario, the context transfer procedures are activated after the completion of the handover at the lower levels of the protocol stack, therefore by definition $N_{\texttt{lost}} > 0$ and $\Delta T_{\texttt{avail}} < 0$. Such a message flow scenario, definitely cannot be used to improve QoS.

In the Mobile Initiated scenario (figure 1(b)), in order for the context to be timely available at the nAR, the CTAR message must be sent as soon as possible, and the context transfer must be completed before the tunnel is established between the two access routers. In case of high mobility, the Mobile Initiated scenario shows a high $N_{\texttt{default}}$ value. It is worth noting that the tunnel setup is faster than the context transfer procedure and that the necessary time to establish a tunnel between the ARs could be saved by means of persistent connections.

The access router initiated scenario guarantee that $\Delta T_{\texttt{avail}} > 0$. The anticipation of the context transfer procedure can be delayed to reduce the waste of bandwidth due to the necessity to send the context related information to all the candidate nARs, thus giving the possibility to the pAR to based the procedure on a more refined choice of candidates. A high delay in the context transfer procedure brings to a scenario that is very similar to the mobile initiated one, showing that tradeoff solutions could be considered between a high bandwidth waste for many anticipated context transfers that guarantee high handover performances, and a low bandwidth waste of a delayed context transfer scenario that could lead to handover performance degradation.

5 Conclusions and Remarks

A considerable number of network services characterized by long lived sessions show a strong need for transparent procedures to transfer context information between network access points. The context transfer must be efficient to support low-latency and real-time application.

In this paper we made a performance analysis of context transfer protocols, comparing three scenarios differentiated on the basis of the trigger mechanism in use to activate the context transfer procedures. Our analysis points out that for small context data a mobile initiated procedure guarantees a good performance also for clients showing high mobility. We also explain how predictive mechanisms, reduce the cost of handovers (in terms of number of lost packets and of packets processed as default), though requiring more bandwidth than dummy or mobile initiated solutions. Protocols optimizations can be introduced to reduce the number of CTD messages sent to candidate nARs.

Acknowledgements

The work of Novella Bartolini has been partially funded by the WEB-MINDS project supported by the Italian MIUR under the FIRB program. The work of Emiliano Casalicchio and Salvatore Tucci has been partially funded by the PERF project supported by the Italian MIUR under the FIRB program.

References

1. Bartolini, N., Campegiani, P., Casalicchio, E.: A Performance Model Of Context Transfer Protocol. Dipartimento di Informatica Sistemi e Produzione, University of Tor Vergata, Roma. (2004)
2. Hinden, R., Deering, S.L.: IPv6 Addressing Architecture. RFC 2373, (1998)
3. Hinder, R., Deering, S.: IP version 6 Addressing Architecture. IETF Internet Draft, (2003)
4. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. IETF Mobile IP Working Group RFC 3775, (2004)
5. Kempf, J. Ed.: Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network. Network Working Group, RFC3374, (2002)
6. Koodli R.: Fast Handovers for Mobile IPv6. IETF Mobile IP Working Group Internet-Draft, (2004)
7. Loughney J. et al. Context Transfer Protocol. IETF Seamoby WG Internet-Draft, (2004)
8. Nartel T. et al.: Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, (1998)
9. Thomson, S., Narten, T.: IPv6 Stateless Address Autoconfiguration. RFC 2462, (1998)