The Connectivity Control Framework: Enabling Session Continuity in Multi-Domain Environments

Michelle Wetterwald
HeNetBot
Sophia Antipolis, France
e-mail: michelle.wetterwald@henetbot.fr

Christian Bonnet
EURECOM
Sophia Antipolis, France
e-mail: christian.bonnet@eurecom.fr

***Abstract*— Last decades have witnessed a massive evolution of mobile communications. When no agreement between the network providers exists, changing the attached network still means breaking the session and relying on the application to recover the lost data. A large set of mobility solutions has been proposed, which impact the network architecture or the applications communications methods. To cope with this issue, this paper presents an innovative technological framework, which applies changes to the terminal only and ensures the continuity of the session when roaming through independent wireless access networks. This framework is based on abstract interfaces hiding the specificities of technologies, a shared knowledge base constantly improved by system learning, and generic service enablers dedicated to specific connectivity tasks, such as a socket session handler. A simulated model based on a heterogeneous wireless playground is used to prove the benefits of this distributed system, which is easily suitable for deployment.**

Keywords-heterogeneous networks; IEEE 802.21; session continuity; multi-domain mobility; generic service enablers; device abstraction; autonomous systems.

I. INTRODUCTION

The evolution of mobile communications has generated new challenges for the design of the connectivity functions in future terminals. The trend has been the conception of multimode devices, with an increasing number of interfaces, and able to connect to any available network. A multimode Mobile Terminal (MT), such as a laptop, smartphone, tablet or car device, is equipped with several network interfaces and able to support communications through one or several of these interfaces at a given time. In parallel, users' requirements in terms of Quality of Experience (QoE) have been soaring, triggering a massive effort from network designers and the conception of devices more and more complex. Because of the turn up of various wireless standards and technologies with different properties, mobile networks have become heterogeneous, incorporating several types of access technologies under the same administrative domain. These accesses provide different connectivity characteristics to the user applications and protocols and require additional adaptability and system control at higher levels of the protocol stack to allow seamless roaming through the different accesses. Roaming and mobility across heterogeneous networks are thus part of the critical operations under study in mobile communications. Currently, when no federation exists between two mobile network providers and no mobility-specific mechanism is deployed in the wireless network, roaming very often means that the session hosting the running application is broken and must be restarted manually, at the cost of lost data, except if the application is designed to recover by itself.

Most of the existing popular user applications use Transmission Control Protocol (TCP) as basis or for the control of their data transfer. TCP has been designed as a stationary protocol, so when the identifier of one of its endpoints, i.e., its Internet Protocol (IP) address and socket port, changes, the TCP connection fails and is terminated. Some of the applications freeze or stop their execution while others are set to establish a new TCP connection and resume their activity. But at the end, it all depends on the way the application itself was developed.

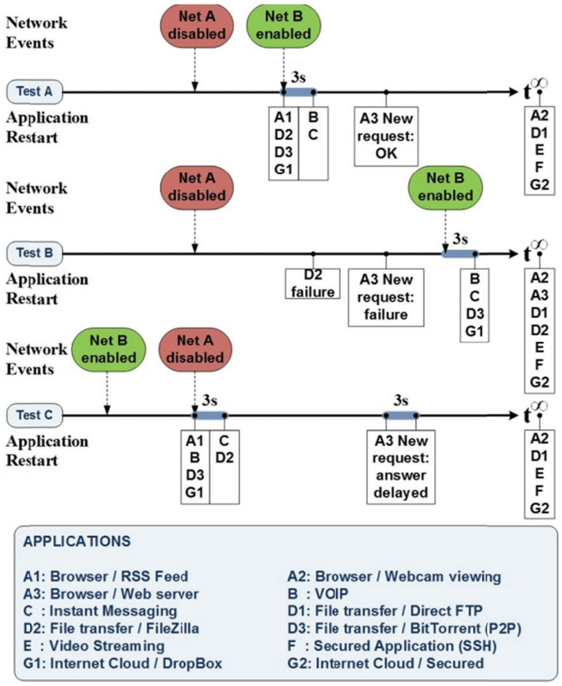


Figure 1. Application Recovery Timeline

Fig. 1 shows the result of experiments that have been performed using a real terminal and applications to frame the issue of application session continuity. The terminal was equipped with two network interfaces accessing independent networks. For each test, the application was started on NetA, then Net A was disabled and NetB started with different delays, as shown in the figure. We could observe that, while some of the popular user applications tested are able to recover by themselves thanks to their smart design, others are frozen or completely stopped, as shown with an infinite time recovery on the right of the figure.

So, the primary objective of this paper is to tackle the problem of the session failure when changing to an access network that does not support mobility. The result will allow an individual using a mobile device to roam seamlessly across non-federated heterogeneous wireless environments. Such environments can be a mobile operator network, a campus hotspot, a road operator communication network or the user private home network.

Several networking techniques such as Media Independent Services, mobility management, or autonomous systems can contribute to achieve our objective, but none of them provides the solution on its own. By enhancing and combining efficiently these mechanisms, the target scheme involves a strong level of cross-layer design and enables many generic services such as handovers, broadcast services, session mobility, battery saving or security.

In this contribution, we propose an innovative framework to resolve at MT level the problem of session continuity between independent domains and leave the network totally unaffected. The rest of the paper is organised as follows. Section II discusses the existing technologies available to address this type of issue, analysing their potential contributions to an integrated framework and their limitations. After this, in Section III, we propose our integrated framework and describe its internal components. This is followed in Section IV by its evaluation with a simulation model implementing the proposed system, and includes its main results. The document is closed in Section V with an assessment of the contribution and the indication of direction for future research topics.

II. ANALYSING EXISTING TECHNOLOGIES

In this section, the existing technologies and challenges lying in the path of the target architecture are identified and reviewed.

A. Media Independent Services

Operating multimode devices in heterogeneous networks can become very complex if each access technology has to be controlled directly and separately by the upper layer entities. This has led to the emergence of a strategy based on a shared abstraction layer above the access layer. In this direction, the IEEE 802.21 standard proposes three different Media Independent Handover (MIH) Services [1], which offer to the upper layer management protocols some generic triggers, information acquisition and the tools needed to perform handovers. The Event Service (MIES) provides the framework needed to manage the classification, filtering and triggering of network events, and to report dynamically the status of the different links. The Command Service (MICS) allows the upper layer management entities to control the behaviour of the links. The Information Service (MIIS) is distributes the topology-related information and policies from a repository located in the network. They result in a cross-layer architecture where the Media Independent Handover Function (MIHF) operates as a relay between the media-specific Link layer entities and the media-agnostic upper layer entities, or MIH-Users. In existing solutions, the MIH-User is represented by a Connection Manager (CMGR) whose main role is to decide which path if best suited to reach the application server or the Correspondent Node (CN) located across the Internet [2].

Currently, the IEEE 802.21 standard provides valuable mechanisms to control the network interfaces of a multimode terminal in a media-independent and abstracted way. However, it involves a few strong limitations. It currently only enables handover services and deals exclusively with the control of wireless network interfaces. It does not consider the information from other devices, such as battery consumption or positioning, in the terminal. It thus offers the possibility to be developed to support an extended set of services and devices in the terminal. This extension will be a main axis for the design of the target solution.

B. Handling Mobility

The most recent mobile devices are expected to be usable while walking on the streets, carried in road vehicles or even in fast speed trains. However, when a device moves out of its original routing area, it cannot continue using the same IP address and the executing session is broken. Incoming packets are still forwarded along the former route and are not able to reach the mobile anymore. To solve this problem, the IETF groups address the issue of mobility at various levels of the protocol stack. The solution coverage is wide spread as well: at device, transport, session, application, or even more recently, at flow level. The objective is to design protocols able to survive the change of the terminal environment context or discontinuities of its connectivity. A large set of mechanisms and protocols has been proposed to solve this issue. Mobile IP and its enhancements, Fast Mobile IP or Proxy Mobile IP (PMIP), operate at the network layer level. Other protocols like mobile Stream Control Transmission Protocol (mSCTP) or Session Initiation Protocol (SIP) address the transport layer level or above [3]. All these solutions thus affect the network or transport layer. They depend on control entities located in the network that must be owned and maintained by specific organizations. They most often infer heavy changes to the network architecture, including in the anchor point, or to the communication interface at the application, and thus face strong unwillingness for their deployment. For their part, the cellular systems handle mobility with 3rd Generation Partnership Project (3GPP) proprietary protocols and procedures [4], sometimes adapted from the previous ones.

Beside these continuous mechanisms, an interesting technique named Delay Tolerant Networks (DTN) allows the mobile nodes to survive long connectivity disruptions.

Intermittent connectivity is overcome by using store-and-forward message switching [5]. Whole or pieces of a specific message are moved between persistent storage nodes (called DTN nodes), which buffer the message pieces for long periods of time until they are able to forward them to the next DTN node. This functionality is provided by an end-to-end message-oriented overlay, called the “Bundle layer”, which is inserted between the application and the transport layer. Since the DTN nodes terminate transport protocols at the Bundle layer, it makes this architecture tolerant to delay and connectivity problems. However, these techniques are over-sized compared to the requirements of a short handover

C. Automating the System

Recent conceptual studies of future network architectures introduce a totally new cognitive plane, where the environment is sensed and observed, leading to the acquisition of knowledge. This is exploited in a novel capability of self-management [6]. The system operates by undertaking intelligent control loops [7]. It senses its operating environment, works with models that analyse its own behaviour in that environment, and, based on existing policies and learned knowledge, derives the appropriate actions to adapt and change the environment, its own state or its operation. A basic knowledge source is installed at setup and further enhanced by self-learning in an evolutionary process through progressive steps. These self-management architectures have been designed in a layered fashion with a hierarchy of decision modules monitoring the information retrieved from sensors and actively coordinating the action of executors, while maintaining a common cross-layer knowledge base. They are currently used for cognitive radio or the management of network infrastructures. By mirroring their functionalities, it sounds interesting to apply the same concept to the self-configuration and self-healing of the MT connectivity in order to optimize its operation.

The individual technologies that have been analysed in this section can be adapted to obtain a better optimized solution. Their efficiency can be improved by combining them into a single framework. Their common factor is the mobile terminal, which is the only node that the end user controls. Accordingly, in the remainder of this study, an innovative approach has been adopted, choosing to apply the designed changes to the mobile terminal only and leaving the network totally unaffected. The connectivity has to be maintained efficiently while remaining transparent to the applications. The system should capitalize on the layered architecture introduced in autonomous systems.

III. THE CONNECTIVITY CONTROL FRAMEWORK

Accordingly, this section proposes a solution based on a cross-layer architecture that leverages the optimization of dedicated generic services. These services operate in close relationship with an abstraction layer, which hides and takes care of the specificities of the embedded devices. Different services, such as access network selection, connectivity and mobility management, and application session management are combined and enhanced to reach the objective of seamless connectivity. Moreover, the new cognitive capabilities of autonomous systems are involved to bring autonomy to the roaming and support a faster decentralized operation.

Following the requirements defined above, the resulting layered system, the Connectivity Control Framework (CCF), pictured in Fig. 2, modifies only the MT, leaving the network unchanged. It revolves around three main principles that guarantee a simple and flexible architecture, and which could be summed up in a simple modification of the terminal operating system. The first principle is to share the knowledge about the terminal context and its environment in a cross-layer fashion. This is achieved by the Cross-Layer Agent (CLA). It stores the configuration, policies and status of the whole framework in a Local Information Base (LIB). The second principle is to hide the heterogeneity and diversity of the internal devices and access networks behind an abstract interface, which facilitates a range of services wider than handover management. This is achieved by the Media Independent Services Function (MISF) and the Link Interfaces, inspired from the MIH model. The third principle is to provide coordinated generic service enablers that can take care of dedicated operations. They ensure the terminal seamless and optimized connectivity and its operational behaviour, coping with dynamic changes and events in the network environment, while preserving the application data transfer continuity. This is achieved by the Network Access Generic Service Enabler (NAGSE), the Mobility Generic Service Enabler (MGSE) and the Session Generic Service Enabler (SGSE). To enhance their efficiency, the Cognitive Manager (CM) has been introduced. It coordinates autonomously the actions of the GSEs and relies on human interaction (component User Interactions Application (UIA)) only when the level of confidence of its self-management algorithms is too low. The GSEs and the CM are integrated in the Connectivity Agent (CA) sub-system.

The components defined in the CCF combine their individual actions in order to bring the whole framework to its expected level of resilience and efficiency.

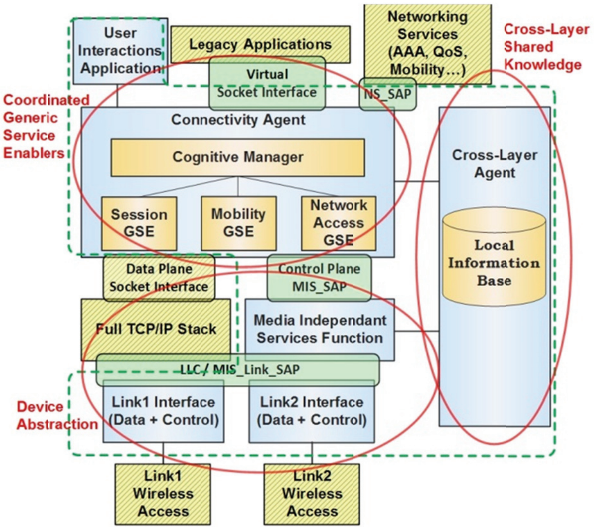


Figure 2. Global architecture of the CCF

A. *Shared Knowledge*

Implementing the first CCF concept, the CLA gathers the data from the devices and Link Layer technologies and from the upper layers, aggregates them in the LIB and provides them on request when needed to the other CCF components.

The cross-layer approach adopted here is a hybrid of two common solutions, in order to integrate their benefits. A cross-layer engine, the CLA, is introduced that works as a local Information Server. It manages a local storage, the LIB, making it accessible to the other components in the framework, namely the MISF and the sub-components of the CA, while preserving its integrity. In parallel, direct interactions between the adjacent components (CM, GSEs, MISF, and Link Interfaces) are maintained to transfer related events and commands, according to the layered model. This scheme distributes the complexity and ensures a quick response of the overall framework to changes in the external environment.

The LIB is the shared knowledge source for the whole system. It contains all the data relevant for an optimized operation of the framework. These data are classified in three types: (i) pre-defined information stored at configuration time, either by the user or by accessing remote databases at the network operator servers (e.g., MIIS) or in the cloud, (ii) status information about the mobile and its environment reported by the other CCF components, (iii) policies and utility functions resulting from the learning process.

B. *Device Abstraction*

The MISF is an enhanced abstraction layer responsible for dealing with the wireless multimodality of the terminal. It is a key component of the connectivity optimization process, as it also provides the means for the abstracted interaction between the radio access and the upper layers. It is based on the IEEE 802.21 MIH model, keeping only its local components, but is not restricted to handover; it fully manages the wireless accesses and the other devices in the terminal, hiding their specificities to the upper layers. It provides a whole set of additional services, including system statistics and status retrieving, resource configuration to comply with a certain level of Quality of Service (QoS), setting and getting identities, handling power sources, positioning, etc. Moreover, it also provides an abstract interface to the CLA component, directly forwarding to the local storage the network or device information received, and contributing to the system learning. The MISF is later able to retrieve the link parameters when requested to issue a command to the lower layers, avoiding that the upper layers get involved with the device details.

The Link Interface components make the link between the MISF and the technology drivers. There is one Link Interface per type of device, completely specific to its implementation. Its main function is to translate the MIS commands and forward them to their target destination. It acts as the endpoint for parameters retrieval from the device. Its location at the edge of the CCF minimizes the overall energy and processing power consumed by the framework. Before generating events, it smooths the values of retrieved dynamic parameters and applies hysteresis thresholds to avoid unnecessary and too frequent reactions. When related to the mobile connectivity, internal devices (positioning systems, power supplies or other sensors) can be integrated and coordinated through the CCF and the MISF, provided the availability of a Link Interface component that translates the abstract MIS primitives into the corresponding set of commands.

C. *Coordinated Generic Service Enablers*

The Generic Service Enablers, or GSEs, are the key elements of this framework. They allow the legacy services to benefit from the technology-agnostic framework. They complement at service level the abstraction introduced by the MISF. These functional blocks are called generic because each of them provides a set of specialized functionalities; they take care of the specificities of the applications and legacy Network Services (NS). They act as MIS-Users and hide the MISF interface to their own users. They can query the LIB for aggregated relevant cross-layer metrics and to provide them to the upper layer services.

The NAGSE deals with aspects related to the monitoring of the networks availability, learning the characteristics of the unknown accesses and selecting the best access network by running its algorithm on a set of parameters retrieved from the CLA. The reader is referred to [8] for more details on its functionality.

The MGSE is the most commonly developed part of the CA. Its role is generally included in the CMGR. It takes care of connectivity related services, including network interfaces and link management, networking aspects, reception and filtering of network and device events, keeping track of current location and connectivity. The MGSE enables the capability to be connected to different types of networks using abstracted processes and mechanisms. Moreover, it smartly filters and dynamically reports changes of the MT context, whether internal or in the external network environment. The process of the MGSE is completely independent from the existing mobility protocols, such as Mobile IP, that may be available in the network and/or the terminal and would run as part of the NS. It thus provides an enhancement to these mechanisms.

The SGSE deals with aspects related to the management of the data sessions opened by the applications. It also takes care of the availability of resources according to the application QoS requirements. For each user application, two related addresses are used. A Personal Address (PA) is attributed to the application when it starts. This address is kept identical throughout the whole session [9]. The Local IP Address (LA) depends on the user network location. It is the global address seen by the external network nodes. In case Mobile IP is used, the LA is equal to the Care of Address (CoA). In the data plane, the SGSE performs the address translation between the PA seen by the application and the LA seen by the network. As will be demonstrated in the validation part in Section IV, this action has a very low impact of the performance of the whole system. For the duration of handovers, it executes a buffering mechanism inspired from DTN techniques, storing the packets received from the application in a temporary local storage until a new connection is safely established. It re-starts the TCP connectivity when it has been broken due to the change of network connection. When an application starts, it opens a socket on a Virtual Socket interface, providing the address and port number of the destination. From the application, this interface is seen as the standard socket Application Programming Interface (API) unchanged. Then, the SGSE opens a real socket on the TCP/IP stack, with the same properties as the virtual one, providing the LA as source address. During an inter-domain handover, the SGSE copes with the break of the session by automatically re-establishing a fresh one through the new access network, using the same parameters. It then updates the binding between the PA and the new LA in order to transfer the flow to the new session. In case TCP is used, it makes sure that the TCP congestion window is set to the same value as it was with the previous session before it failed, in order to avoid the slow start mechanism and reduce the impact of the handover to the local application.

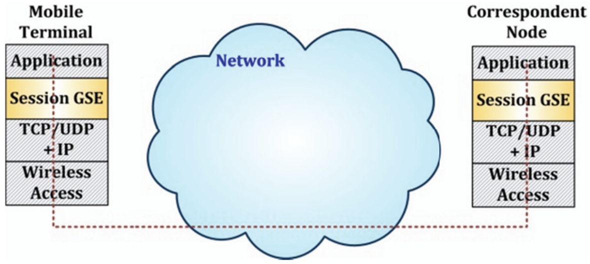


Figure 3. Communication with the correspondent node

The session continuity is ensured at the CN as well, whether it is a server or another mobile terminal, and despite the break of the TCP connection. A simplified version of the CCF, including only the SGSE, is present in the CN and executes the same address translation and packets buffering as in the MT. It is thus able to hide the change to its endpoint of the application. This solution is scalable, as it does not require any intermediate node. The mobility is handled above the transport layer; it is compatible with existing mechanism and protocols, thus opening the path to fast acceptability. The SGSE brings to the framework the mechanisms that allow it to recover when a terminal movement has endangered the operation of a running application.

The GSEs are directly interacting with the MISF, each of them dedicated to a specific role. They need to be coordinated to provide an integrated autonomic behaviour. The CM plays the role of the system controller, orchestrating the self-management functions in the MT to increase the level of efficiency of the global framework. It is assisted by the UIA, a simple user interface to the human owner of the terminal. The CM obtains information and triggers actions from the GSEs. It coordinates their actions according to its own state and the events received. It makes the operational decisions using the knowledge source in the CLA, and triggers their execution. The CM brings to the framework the capability to operate in a smart and autonomous manner, hiding the complexity of maintaining the network connectivity from the mobile user. The overall terminal operation benefits from increased robustness, adaptability to internal or external events and enhanced effectiveness.

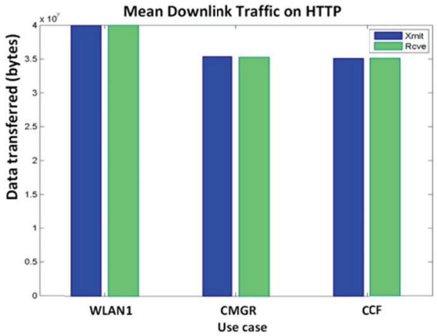
The UIA establishes the link with the mobile user and is the component which allows him to control the operation of his mobile according to his needs and requirements. It obtains the user preferences at configuration time or is used to validate the CCF decisions at runtime, i.e., every time some human-originated information is needed. Such an interaction makes sense and is expected to become less tedious with the apparition of more intuitive user interfaces based on voice rather than screen pop-ups.

IV. MODEL VALIDATION AND RESULTS

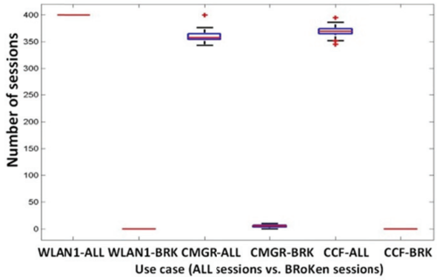
This section aims at validating the chosen technological scheme. The test displayed in Fig. 1 could be reproduced with a simulation model that was developed with the objective to demonstrate the capabilities and the benefits of the framework approach. The model focuses on adding the CCF components to a wireless multimode terminal and moving randomly that terminal in a sample wireless network. By doing so, it evaluates the impact of the framework in the context of the scenario described above, using a selection of common applications (file transfer, Web browsing). The framework here is assessed on its efficiency rather than traffic throughput perspective. The evaluation criteria are the number of data bytes that did not arrive at destination, the recovery from broken TCP sessions and the time between two handovers to control the Ping-Pong effect between the two access networks.

The prototype has been developed as a simulation running under the OMNET++ tool [10]. All the components of the CCF framework shown in Fig. 2 are implemented and tested in the simulator. A small part of the features are streamlined due to the wide range of services described in Section III. The following statistics are collected during each simulation run: number of bytes transmitted and received by the applications, number of TCP connections opened / broken during the test, number of handovers (HO) and time between two handovers, connection time on each technology and in total, usage of the DTN buffers and DTN process duration, time when the last packet was received by the application in the mobile. Each test was performed under steady state conditions, with a fixed duration of 2000s of simulation time and is reproduced 50 times to allow a reasonable simulation time, while providing sufficient data for results aggregation, as shown in Fig. 4b and Fig. 5b.

The results obtained with the CCF prototype-enabled terminal moving randomly across the playground are compared with two other similar use cases that do not involve any change in the network and transport layers or in the network infrastructure: standard multimode terminal with no connectivity control and mobile terminal equipped with a CMGR equivalent to current smartphones. The results obtained from the simulation runs, shown in Fig. 4 and Fig. 5, focus on the critical issue of the TCP session continuity and recovery.

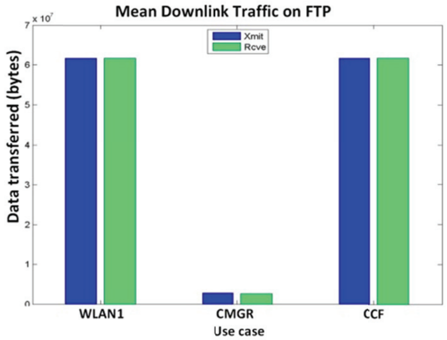


HTTP Session: data transfer DL

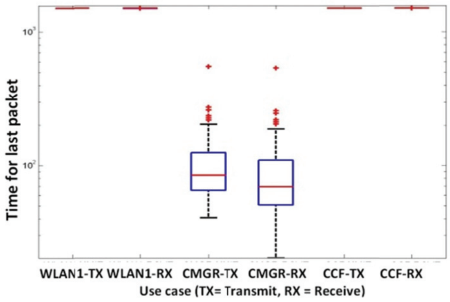


HTTP Session: broken sessions

Figure 4. Simulation results for web browsing



FTP Session: data transfer DL



FTP Session: Last packet timestamp

Figure 5. Simulation results for file transfer

When executing the web browsing application, no session is broken anymore midway to its completion (CCF-BRK box in Fig. 4b), as happened with the CMGR (CMGR-BRK box in Fig. 4b). The amount of traffic transferred is slightly reduced, because the application is interactive and a larger part of the data traffic is transferred through the safer cellular access which, on the other hand, offers a reduced bandwidth.

These results show that when executing a file transfer application, the virtual TCP connection is not broken anymore. With the CMGR (result in the middle of each graph), the traffic exchange stopped after the first handover and resulted in a very low quantity of data exchanged. With the CCF, the application can complete its task until the end and transfer all its packets, as in the reference case on the left. Some of them require the additional support of the buffering queue technique, as can be seen in Table 1.

Table 1 shows the usage of the buffer queue in the SGSE. A few packets are saved during the connectivity break and sent when it is complete, which fully prevents packet loss. We could observe that their number during a specific HO remains very low, at a level quite acceptable compared to the amount of memory available in a mobile terminal.

TABLE I. MEASURES FOR BUFFERING MECHANISM

Parameter	**Use Case**	
	FTP	***HTTP***
Average Number of Handovers	15.65	15.47
Total packets queued - max	36.00	10.00
Total packets queued - min	2.00	0.00
Max packets queued during 1 HO	3.00	1.00
Min packets queued during 1 HO	2.00	0.00

Finally, statistics have been collected that demonstrate the low fingerprint of the CCF operation on the MT processing power and the usage of the buffering mechanism, as it is measured smaller than 0.2%. On a specific test where the MT executes the file transfer application and performs 12 handovers, only 1233 events (over 761839 for the full simulation run) did involve one of the CCF components. It represents less than 0.2% of the total operations.

V. CONCLUSION AND FUTURE WORK

This paper has presented the design and evaluation of a cross-layer, integrated and coordinated framework, the Connectivity Control Framework, whose target is to ensure session continuity across independent wireless networks. The CCF is restricted to the mobile terminal and has no impact on the mobile network infrastructure, while maintaining full compatibility with existing networking standard. Its layout is based on three main principles: (i) a cross-layer agent, which maintains and shares the knowledge acquired by the other components of the framework; (ii) an abstraction layer, which hides the network specificities to the rest of the framework, including as well the support of other hardware devices such as positioning systems or diverse sensors; (iii) coordinated generic service enablers responsible of dedicated tasks and taking care of the various functions necessary to handle the terminal connectivity. To assess its benefits, a simulation model has been developed, that compares the framework behaviours in a testing heterogeneous environment to other solutions that do not affect the network. The concept of keeping the changes in the mobile device distributes the effort in the global system, reduces the risk of bottleneck functions in the network and improves its scalability. No additional network entity has to be deployed and maintained by the operators, the system installation and configuration are simplified. This study has investigated comprehensively the distribution of functions and the integration and coordination of the system proposed. For each individual component, existing solutions that satisfied the main requirements have been selected and necessary enhancements described. A future continuation of this topic will perform a more precise analysis and definition of the generic service enablers introduced in the CCF.

REFERENCES

[1] E. Piri and K. Pentikousis, “IEEE 802.21”, the Internet Protocol Journal, Volume 12, No.2, June 2009, pp. 7-27.

[2] M. Kassar, B. Kervella, and G. Pujolle, “An overview of vertical handover decision strategies in heterogeneous wireless networks”, Computer Communications, Volume 31, Issue 10, June 2008, pp 2607-2620.

[3] G. Lampropoulos, N. Passas, L. Merakos, and A. Kaloxylos, "Handover management architectures in integrated WLAN/cellular networks", IEEE Communications Surveys & Tutorials, February 2006, pp 30-44.

[4] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", Release 10, June 2012, available at ftp://ftp.3gpp.org, last accessed on 05/2014.

[5] F. Warthman, "Delay-Tolerant Networks (DTNs): A Tutorial; v1.1", Wartham Associates, 2003.

[6] J. Larsen, "Cognitive Systems", tutorial presented at IEEE Workshop on Machine Learning for Signal Processing, Cancun, Mexico, October 2008.

[7] IBM Corporation. “An Architectural Blueprint for Autonomic Computing,” White Paper, 4th Edition, June 2006.

[8] M. Wetterwald and C. Bonnet, "Devices and wireless interface control in vehicular communications: an autonomous approach", Ambi-sys 2013 , Athens, March 2013, pp 91-103.

[9] R. Bolla, R. Rapuzzi, and M. Repetto, “An Integrated Mobility Framework for Pervasive Communications”, Proc. of IEEE Globecom 2009 Next-Generation Networking and Internet Symposium (GC’09 NGNI), Honolulu, Hawaii, USA, Nov.-Dec. 2009, pp 1-6.

[10] OMNET++ simulator, available at http://www.omnetpp.org/, last accessed on 05/ 2014.