2012-ENST-0064

EDITE - ED 130

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique et Réseaux »

présentée et soutenue publiquement par

Michelle WETTERWALD

le 12 novembre 2012

CONTROLE DE TERMINAUX MULTIMODES AUTONOMES DANS DES ENVIRONNEMENTS SANS-FIL HETEROGENES ET INDEPENDANTS

Directeur de thèse : **Prof. Christian BONNET**

Jury

M. Rui L. AGUIAR, Prof., Universidade de Aveiro, Portugal	Rapporteur
M. Jean-Marie BONNIN, Prof., Telecom Bretagne, France	Rapporteur
M. Chadi BARAKAT, HDR, INRIA, France	Examinateur
M. Jeremie LEGUAY, PhD, THALES, France	Examinateur
M. Pietro MICHIARDI, MdC, EURECOM, France	Examinateur

THÈSE

TELECOM ParisTech
Ecole de l'Institut Télécom - membre de ParisTech
46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2012-ENST-0064

EDITE - ED 130

Doctorate ParisTech

THESIS

in partial fulfillment of the requirements
for the degree of doctor from

TELECOM ParisTech

Specialization « Computer Science and Networking »

presented and publicly defended by

Michelle WETTERWALD

on November 12, 2012

CONTROL OF AUTONOMOUS MULTIMODE TERMINALS IN HETEROGENEOUS AND INDEPENDENT WIRELESS ENVIRONMENTS

Thesis Supervisor : **Prof. Christian BONNET**

Jury

M. Rui L. AGUIAR, Prof., Universidade de Aveiro, Portugal	Reporter
M. Jean-Marie BONNIN, Prof., Telecom Bretagne, France	Reporter
M. Chadi BARAKAT, HDR, INRIA, France	Examiner
M. Jeremie LEGUAY, PhD, THALES, France	Examiner
M. Pietro MICHIARDI, MdC, EURECOM, France	Examiner

THÈSE

TELECOM ParisTech
Ecole de l'Institut Télécom - membre de ParisTech
46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

REMERCIEMENTS

Tout d'abord, je tiens à exprimer ma profonde gratitude à mon Directeur de thèse Christian Bonnet pour sa remarquable supervision et ses encouragements tout au long de ma thèse. Je tiens à le remercier pour la liberté et l'orientation technique qu'il m'a prodiguées dans mes recherches. Sans sa compréhension et ses conseils avisés, je n'aurais pu mener ce travail à son terme. Ce fut un réel plaisir et une expérience fructueuse de travailler avec Christian.

Je voudrais ensuite remercier Telemaco Melia pour avoir créé et dirigé le projet MEDIEVAL qui m’a permis de trouver le terreau nécessaire à cette recherche. Je remercie tous les membres du groupe du WP3 pour l’excellente coopération qui a entouré notre activité. Nos derniers résultats témoignent du succès de notre groupe. Je suis particulièrement reconnaissante envers Daniel Corujo et Davide Chiarotto pour l’aide qu’ils m’ont apportée dans ce travail.

Je tiens aussi à remercier mes collègues Akis, Pavlos et Lionel pour m’avoir aidé à finaliser ma thèse. Merci à Gwenaëlle pour son aide et son soutien indéfectible pendant toute la durée de la thèse.

Je voudrais remercier l’équipe de SpiderOak pour m’avoir offert tous ces Go en ligne qui m’ont permis de travailler de façon sûre et sereine.

Finalement, je voudrais rendre hommage à ma famille qui m’a soutenue dans ce projet et a fait d’énormes efforts pour le rendre réalisable. Merci à mes enfants Noémie et Damien pour l’aide précieuse qu’ils m’ont apportée dans la relecture et la finalisation de ce mémoire, si loin de leur activité habituelle. Je remercie Alexis pour avoir partagé ses heures d’études avec moi.

.

ABSTRACT

Recent years have witnessed a massive evolution of mobile communications. Issues related to the device mobility have been thoroughly addressed inside operators' domains, but 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. In parallel, it is hardly possible for a mobile user to control the connectivity of his terminal. The objective of this thesis is to present the concept of an innovative technological framework for the autonomous control of multimode terminals in heterogeneous and non-federated wireless environments. The aim is to enable a self-configuring terminal to connect and roam seamlessly across independent networks as long as it owns sufficient security credentials, while respecting its user's choices and preferences. The target scheme involves abstraction and cross-layer mechanisms. It takes into account constraints based on heterogeneous wireless systems, autonomous architectures and enables generic services such as smart access network selection, connectivity and session management. This scheme applies to the mobile terminal only, with mechanisms independent of the access network infrastructure. The thesis analyses how existing technologies are enhanced and combined with new features to achieve this objective and gives a description of the overall concept and of its implementation. A simulated model is used to assess the validity of the proposed framework. Diverse applications to real systems and projects that implemented the components of this framework are presented, highlighting the lightness, generality and key benefits of the concept.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

Readers can find here the abbreviations and acronyms used throughout the thesis. The meaning of an acronym is usually indicated once, when it first occurs in the text. In some case, it can be repeated to facilitate the readers.

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AAA	Authentication, Authorization and Accounting
AHP	Analytic Hierarchy Process
AMS	Address Management System
ANDSF	Access Network Discovery and Selection Function
AP	Access Point
API	Application Programming Interface
APN	Access Point Name
AR	Access Router
AS	Autonomic Systems
AT	ATtention (modem)
B3G	Beyond-3G
BCH	Broadcast Channel
BER	Block Error Rate
BS	Base Station
CA	Connectivity Agent
CAN	Content Aware Networks
CAS	Connection Abstraction System
CDMA	Code Division Multiple Access
CoA	Care of Address
CCF	Connectivity Control Framework
CLA	Cross-Layer Agent
CM	Cognitive Manager
CMgr	Connection Manager
CN	Correspondent Node
CP	Communication Profile

CPU	Central Processing Unit
CR	Cognitive Radio
DAB	Digital Audio Broadcasting
DHCP	Dynamic Host Configuration Protocol
DMM	Distributed Mobility Management
DNS	Domain Name System
DTMA	Disruption Tolerant Mobility Architecture
DTN	Delay (or Disruption) Tolerant Network
DVB	Digital Video Broadcast
DVB-SH	Digital Video Broadcasting for Satellite Handheld
eMBMS	evolved Multimedia Broadcast Multicast Service
EnvA	Environment Address,
EPS	Evolved Packet System
ESM	EPS Session Management
ETSI	European Telecommunications Standards Institute
EUI-64	64-bit Extended Unique Identifier
FIFO	First In, First-out
FOT	Field Operational Trial
FTP	File Transfer Protocol
GN	GeoNetworking
GNC	GeoNetworking Control
GNSS	Global Navigation Satellite System
GP	Generic Profile
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSE	Generic Service Enabler
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HA	Home Agent
HeNodeB	Home eNodeB
HIP	Host Identity Protocol
HMIP	Hierarchical Mobile IP
HO	Handover
HoA	Home Address

HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IAM	Interface Autonomic Manager
I2V	Infrastructure to Vehicle
ID (Id)	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol (Version 4: IPv4; Version 6 : IPv6)
ISO	International Standards Organisation
ITS	Intelligent Transport Systems
ITSS	ITS Station
L2	Layer 2
L3	Layer 3
LDM	Local Dynamic Map
LIB	Local Information Base
LIPA	Local IP Access
LLC	Logical Link Control
LTE	Long Term Evolution
LTE-A	LTE-Advanced
MAC	Media Access Control
MADM	Multi-Attribute Decision Making
MAP	Mobility Anchor Point
MBMS	Multimedia Broadcast/Multicast Service
MDP	Markov Decision Process
MGSE	Mobility GSE
MIB	Management Information Base
MICF	Media Independent Control Function
MICS	Media Independent Command Service
MIES	Media Independent Event Service
MIF	Multiple Interfaces
MIH	Media Independent Handover

MIHF	Media Independent Handover Function
MIIS	Media Independent Information Service
MIP	Mobile IP
MIS	Media Independent Services
MISF	Media Independent Services Function
MME	Mobility Management Entity
MSN	Microsoft Network, now Windows Live Messenger
MT	Mobile Terminal
NAA	Network Aware Applications
NAGSE	Network Access GSE
NAS	Non Access Stratum
NetLMM	Network-based Localized Mobility Management
NS	Networking Services (components)
N&T	Networking and Transport (ITS layer)
OAM	Orchestrating Autonomic Manager
OMA	Open Mobile Alliance
OSI	Open Systems Interconnection
P2P	Peer to Peer
PA	Personal Address
PDA	Personal Digital Assistant
PDN	Public Data Network
PDN-GW	PDN Gateway
PHY	Physical (layer)
PMIP	Proxy Mobile IP
PoA	Point of Attachment
POI	Point of Interest
PoS	Point of Service
PPP	Point-to-Point Protocol
QoE	Quality of Experience
QoS	Quality of Service
RAdv	Router Advertisement
RAL	Radio Access Layer
RAM	Random Access Memory
RAT	Radio Access Technology

RFC	Request for Comments
RFID	Radio Frequency IDentification
RRC	Radio Resource Control
RSS	Received Signal Strength
RSS	Really Simple Syndication
RSU	Road Side Unit
RTP	Real-Time Protocol
RTCP	Real-time Transport Control Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round Trip Time
SAP	Service Access Point
SAW	Simple Additive Weighting
SCTP	Stream Control Transmission Protocol
SFTP	SSH File Transfer Protocol
SGSE	Session GSE
S-GW	Serving Gateway
SIM	Subscriber Identity Module
SINR	Signal to Interference plus Noise Ratio
SIP	Session Initiation Protocol
SIPTO	Selected IP Traffic Offload
SNR	Signal to Noise Ratio
SON	Self-Organizing Network
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TFT	Traffic Flow Template
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TV	Television
UDP	User Datagram Protocol
UE	User Equipment
UIA	User Interactions Application
UMF	Unified Management Framework
UMTS	Universal Mobile Telecommunications System

V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VNAT	Virtual Network Address Translation
VoIP	Voice over IP
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless FIdelity
WLAN	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access
WP	Weighted Products
WPA	Wi-Fi Protected Access
WRAN	Wireless Regional Area Network
WSN	Wireless Sensor Network
XMPP	Extensible Messaging and Presence Protocol

CHAPTER 1 - INTRODUCTION

1.1 Motivation

Recent years have seen the explosion of mobile communications, together with an exponential use of the Internet for all sorts of applications such as voice calls, electronic mails, file transfer, video streaming or Mobile TV. As a consequence, the trend in the mobile technologies has been the conception of multimode devices, with an increasing number of interfaces, at the centre of fixed and mobile or telecom and broadcasting convergence and able to connect to any available network. A ***multimode Mobile Terminal*** (MT) is a computing device, such as a laptop, smartphone, tablet or car device, equipped with several network interfaces and able to support communications through one or several of these interfaces at a given time, according to its usage and its environment. In the same period, the internet has accelerated it evolution rhythm. When 25 years ago, only a few thousand users were connected, nowadays the Internet covers and offers new communication means to billions of people around the world. Users' requirements in terms of Quality of Experience (QoE) are 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 networks under the same administrative domain. These different types of accesses provide different connectivity characteristics to the user applications and protocols. They require additional adaptability and system control at higher levels of the protocol stack to pass through the different accesses. In parallel, people have become aware of climate issues, adding new energy saving constraints to the conception of networks and mobile devices.

In the last few years, the direction adopted in the design of mobile networks, mainly led by operators, has been to keep the MT unchanged and bring all the operational control functions in the network, arguing that intelligence is cheaper at network level than at the node level and that it makes the network easier to manage. This position goes against the spontaneous market tendency to prefer distributed and user-controlled systems. However, the most recent trend is moving back to user-centric approaches [20]. ***Roaming*** and ***mobility*** across heterogeneous networks, in terms of technology but also administration and management, are 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, which is the common case nowadays, roaming means a full change of the network context. Most of the time, the session hosting the running application is broken and must be restarted manually, at the cost of lost data, except if the application is able to recover by itself.

A vast majority of the existing applications, running on a mobile terminal, use the TCP (Transmission Control Protocol) for their data transfer (for example HTTP-based web browsers) or at least to control the correct delivery of the UDP (User Datagram Protocol) data traffic between the end-nodes. TCP has been designed as a stationary protocol, so when the identifier of one of these endpoints, i.e. its IP (Internet Protocol) address and socket port, changes, the TCP connection fails and is terminated. This has been tested during this study on a varied set of common applications, using a mobile terminal with two heterogeneous interfaces. The results show that all the applications signal a connection error. Some of them 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 was developed. The results obtained during these experiments are detailed in Chapter 2.

A parallel issue lies in the ***choice of the access network*** by the multimode terminal. Currently, the behaviour encountered in a smartphone is most of the time pre-defined. It prioritizes a known Wi-Fi hotspot when one is detected, transferring blindly all the data traffic on this access. When no Wi-Fi is available, it transfers all its traffic through the cellular network if one is configured. A laptop with two different accesses active is not able to indicate which one is used by its applications. This may lead to odd or unwanted behaviours. In the case of the laptop, the application performance may be lower than the user expectation. Unknown Wi-Fi network accesses may be briefly suggested when starting a new application while being passenger of a car moving on a highway. The application used may force the usage of a specific access network, or type of access, either for performance, cost or privacy reasons. New access technologies, for vehicular communications for instance, are just about to be deployed, creating an additional level of complexity to the dynamic selection of the optimal access network in a specific context. The current binomial and static solution is thus expected to become rapidly too limited and simplistic compared to the complexity of the mobile environment and to the connectivity constraints foreseen in the near future.

Both issues have the effect of a negative efficiency and Quality of Experience (QoE) perceived by the user. Various mechanisms and protocols are already defined and can be re-used to set up a solution addressing these problems. To handle multimode terminals, the IEEE 802.21 work group [21] has standardized a media independent handoff process common to all the 802 media, and that additionally enables the mobility to and from 3G cellular systems. The standard introduces, above the different media interfaces, an intermediate abstraction layer that provides services to the upper layer entities running in the mobile terminal. As wireless systems became available, the user terminals became nomadic and able to connect anywhere. With the strong reduction of their size, they became truly mobile, adding dynamicity to their connectivity. The issue of intra-domain mobility is being addressed mostly by IETF working groups, at various levels: device, session or even more recently at flow and application level. These standards consider both terminal and network-triggered mobility, yet with a preference for network operation. However, there are scenarios where the mobility cannot be controlled by the operator and should be left to the decision of the mobile user. This is especially true when the user prefers to choose a network different from the default one chosen by the terminal or if one of the available networks is private and has no mobility mechanism deployed. Associated with these techniques, the network access selection algorithms range from simple algorithms, e.g. in cellular networks, where the access with the best signal quality is chosen, to smarter ones which take into account additional requirements from the end user, the application, the mobile terminal context or even network policies provided by functional entities such as the Access Network Discovery and Selection Function (ANDSF) defined in the 3GPP standard. This functional entity is distributed in the network and provides the information about its heterogeneous, but still operator-related, neighbourhood to the mobile terminal. The selection algorithms are usually executed in a Connection Manager, located in the MT, which triggers the switching from the old technology to the new target one. Unfortunately, it is not designed to prevent efficiently the failure of the applications when moving between independent networks. Furthermore, in addition to the traditional data and control planes, the conception of future Internet architectures introduces a totally new cognitive plane, where the environment is sensed and observed, leading to the acquisition of knowledge which is restored in an improved capability of ***autonomous behaviour*** and ***self-management***.

The primary objective of this thesis is to tackle the problem of the application failure when changing the access network without mobility support, especially when it uses the TCP transport protocol, and to enable a fine tuning of his mobile connectivity by the end user. It aims to design a novel global technological scheme, consisting in an innovative framework and the associated protocols. The result will allow an individual using a mobile device to keep the control of his multimode terminal and roam seamlessly across heterogeneous and non-federated wireless environments. Such environments can be a mobile operator network, the local cafe hotspot or the user private home network. Our target is to enable his self-configuring terminal to connect and move freely across wireless environment as long as it owns sufficient security credentials. The study covers topics such as seamless connectivity control of multimode terminals across heterogeneous administrative domains, network access selection and prevention and recovery from broken TCP connections due to network changes. In addition, it considers the recent cognitive capabilities of autonomous systems and targets a light weight execution to avoid impacting too heavily on the battery consumption and processing power of the terminal. Finally, based on the consideration that new techniques and algorithms modifying the existing network entities are difficult to deploy in real life compared to mobile terminal improvements, and that backward compatibility should be achieved in the larger possible extent, this work has adopted an innovative approach to solve the problem, proposing to apply all the necessary changes to the mobile terminal only and leave the network totally unaffected, while being able to comply with any type of network.

Each of the networking techniques mentioned in the previous paragraphs can contribute to achieve our objective, but none of them is sufficient on its own. By combining efficiently their mechanisms to enhance their impact, the target scheme implies a strong level of cross-layer design, taking into account generic constraints based on cognitive wireless systems and enabling generic services such as handovers, broadcast services, session mobility, battery saving or security.

1.2 Key Contributions

The key contributions to the study of control of autonomous multimode terminals in heterogeneous and independent wireless environments proposed in this thesis can be summarized as follows.

- An integrated system has been elaborated to solve the problem of application failure and give control of his terminal to the user. This framework affects the mobile terminal only, leaving the network totally unaffected. It has a small impact on the Correspondent Node in case it is not a mobile terminal. The operational behaviour of this system has been extensively analysed.
- A Cross-Layer Agent has been introduced that aggregates in its Local Information Base

the status and configuration parameters of the different layers involved in the connectivity of the mobile. User preferences and network policies are stored in the same information base. All the parameters can be shared among the various entities of the framework.

- An extended abstraction layer, containing a Media Independent Services Function and device-specific Link Interface components, has been defined. It monitors and controls the behaviour of the heterogeneous network interfaces, hiding their specificities to the upper layer services. It is also able to manage the various devices that may impact the choice of the next access network: positioning system, sensors or battery power management. Moreover, this abstraction layer is able to interact with the cross-layer agent to provide the device status and retrieve directly the configuration parameters related to a specific (action, device) pair.
- The concept of Generic Service Enablers has been proposed. It provides specific and dedicated services to the legacy applications and Networking Services, while interfacing and using the services of the lower layers through the abstraction. To enhance the efficiency of the terminal connectivity, they implement the following functions: network access selection, mobility and connection management and session management.
- A novel Generic Service Enabler for session management has been proposed. It is capable of launching automatically the network authentication through a simple pre-defined HTTP (Hypertext Transfer Protocol) session. Its sub-functions manage the life cycle of an application. It allocates a Personal Address to each application when it starts. It offers to the application a virtual socket API (Application Programming Interface) and performs the mapping of the virtual socket address and port number towards the real output socket IP address and port. Moreover, it executes a short buffering of the packets above the transport layer, in a manner similar to DTN (Disruption Tolerant Network) techniques, while they wait to be sent during a handover.
- A Cognitive Manager, assisted by a User Interactions Application, completes the integrated framework and orchestrates the actions of the Generic Service Enablers, allowing the terminal to function and maintain its connectivity in a quasi-autonomous manner.
- This framework has been implemented in the OMNET++ simulation tool, providing results to evaluate its actual operational feasibility and efficiency versus existing solutions.
- The applicability of parts of this framework has been studied in a very diverse set of real systems and projects, providing theoretical and experimental results of its validity and wide coverage. They have shown that the solution designed is suitable to a diverse set of use cases, from end-user mobile terminal or smartphone to vehicle station device and emergency notification system.

1.3 Thesis Outline

According to the objectives introduced in Section 1.1, this thesis is organized as follows.

Chapter 2 starts with an evaluation of the impact of a TCP broken session on a set of commonly used applications with a real terminal. Next sections investigate several enabling technologies and mechanisms which can contribute to the design of the target architecture. To complete the study, a review of autonomous systems and the way they are applied in the networking domain is performed. Finally, the context of vehicular communications, a specific application domain and textbook case for this study, is described.

Some typical scenarios are elaborated in Chapter 3, showing some use cases which are non-operational with existing mobile terminals. A set of system requirements is derived from these scenarios. Next, a subset of the use cases has been reproduced by simulation to prepare the assessment of the results achieved with the target architecture, validating the legitimacy of the problem.

The proposed solution, consisting in a global layered framework, is presented in Chapter 4. The general description contains a functional system and operational view of its architecture, completed by a summary of the possible interactions with the surrounding network entities. Then, each of the components participating to the framework is described with its main features. This description is completed by the presentation of the interactions between these components, including message sequence charts and interfaces definitions.

Chapter 5 discusses the validation of the defined architecture in a simulation model. A subset of the framework has been implemented and simulated to evaluate the efficiency of the final system. The test scenario is presented and the results obtained are analysed to evaluate the benefits and impacts of the framework.

Chapter 6 describes how the main principles of the system designed were applied to a varied set of projects and situations. The implementations were also used to perform a preliminary evaluation of the functionality of the proposed concept.

Finally, Chapter 7 concludes this work, highlighting the key contributions of the proposed framework and identifying perspectives for future research activities.

CHAPTER 2 - REFERENCE TECHNOLOGIES AND CHALLENGES

In this chapter, the existing technologies and challenges lying in the path of the target architecture are identified and reviewed. The introduction highlighted two major problems in the current operation of mobile terminals. Firstly, when the user of a multimode mobile device moves through independent access networks, his running application is often frozen or stopped because his network identifiers have been changed. Secondly, it is not possible for him to fine tune his own mobile terminal connectivity according to his preferences or the needs of his applications. A revision of the MT functional framework is thus necessary to adapt to these new constraints. Various mechanisms and protocols already exist which can be tailored to set up such an architecture. This chapter starts with an experiment on a set of common applications, assessing and formalizing the TCP connection failure problem on a real terminal. Then, we give an overview of four enabling technologies which are good candidates to be integrated in the target framework and study their interesting features which could be reused, together with their limitations. Finally, the context and the existing model defined for the ITS (Intelligent Transport Systems) communications are presented because they include three access technologies, plus the positioning device, in the vehicle and thus constitute a textbook case for this problem.

2.1 Impact of TCP Broken Session on Common Applications

In order to better evaluate the extent of the problem of the TCP broken connection and how common applications react to the change of IP context, a few tests have been conducted with a dual-mode mobile terminal. Each application was started with an access technology, which was then disconnected while the second technology was connected with a varied delay. The objective was to analyse the effect of a network disconnection in the absence of smart mobility mechanism and validate the effectiveness of the problem of maintaining seamlessly the applications connectivity while roaming through independent and non-operated networks.

2.1.1 Testing Specification

As shown in Table 1, the connection of the second access has been performed according to one of the following chronologies compared to the disconnection of the first access.

1. Test sequence A : connection within a short delay, less than 30s;
2. Test sequence B : connection within a long delay, over 10mn;
3. Test sequence C : connection before the first access is disconnected (“multi-homing” condition).

Testing Sequence A	**Testing Sequence B**	**Testing Sequence C**
Connect Wireless 1 Start application Unplug Wireless 1 Connect Wireless 2 (short delay)	Connect Wireless 1 Start application Unplug Wireless 1 Connect Wireless 2 (long delay)	Connect Wireless 1 Start application Connect Wireless 2 Unplug Wireless 1

Table 1: Testing methodology

Since it is not a convenient task to connect and disconnect the cellular access with an existing mobile phone, while doing similar procedures on the Wi-Fi access, the test has been performed using Ethernet and Wi-Fi technologies on a laptop. We argue that the results are identical, since the wireless property of each network is not really meaningful for the results obtained. What is important is the fact that no mobility management procedure is executed to control and optimize the handover impact.

Table 2 gives the list and properties of the applications that have been evaluated for this study, including a short description of the transport protocols and ports that they use when available.

Type	**Application**	**Description**
Browser	Mozilla Firefox	A browser uses the HTTP protocol to retrieve documents written in HTML and other adequate formats. HTTP runs on TCP, with well-known port 80. Most requests start and are sent over their own specific TCP connection to the HTTP server.
VoIP	Skype	A typical VoIP (Voice over IP) application uses SIP (Session Initiation Protocol, over TCP or UDP, port 5060) for its signalling and RTP (over UDP) for the multimedia stream. Skype is actually a P2P IP telephony network. It uses a TCP connection for the user login and non-standardized protocols for the data, on a port chosen at random (fall back to ports 443 and 80).
Instant Messaging	GTalk (Google)	Google Talk is based on the XMPP protocol

		(eXtensible Messaging and Presence Protocol, originally known as Jabber, streaming of .xml elements) on top of TCP port 5222.
	MSN (Microsoft)	MSN is based on proprietary protocols with well-known TCP ports 3331 and 1863 (for the server).
File transfer	Direct FTP	FTP is a File Transfer Protocol running on top of TCP (port 21 for control flow and port 20 for data flow). It generally establishes one single connection with one single server for the whole duration of the session.
	FileZilla	FileZilla is a client application providing access to FTP servers with a GUI (Graphical User Interface).
	BitTorrent	BitTorrent is a peer-to-peer (P2P) protocol which executes requests for pieces of the original file using several TCP connections (non-official ports) to different machines hosting these pieces.
Video Streaming	YouTube	YouTube uses HTTP/TCP protocols to buffer video into a flash player. On some handsets, the streaming protocol is the RTSP (TCP port 554) protocol for the end-to-end connection, together with RTP for the data (port 5004-5005).
Secured Applications	SSH Client	SSH (Secure SHell) is a network protocol for secure data communications, used in client applications to access shell accounts (replaces Telnet), SSH file transfer (SFTP) or secure copy (SCP). SSH runs on TCP port 22.
Internet Cloud applications	Dropbox	Dropbox runs on HTTP. It maintains one connection for the authentication and one HTTPS (HTTP Secure) connection to send the data.
	Secured Application	This is a commercial file storage application providing a secured access on any type of user terminal.

Table 2: Applications tested

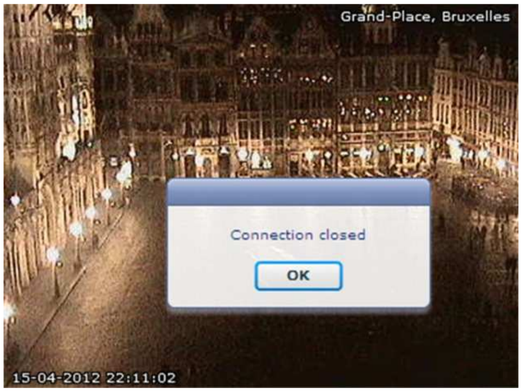
2.1.2 Test Observations

This section describes the observations and, when relevant, the measurements that were made in each evaluation.

Browser

A browser like Firefox allows to access different types of servers, and accordingly, the results are quite varying and depend on the protocol used above HTTP.

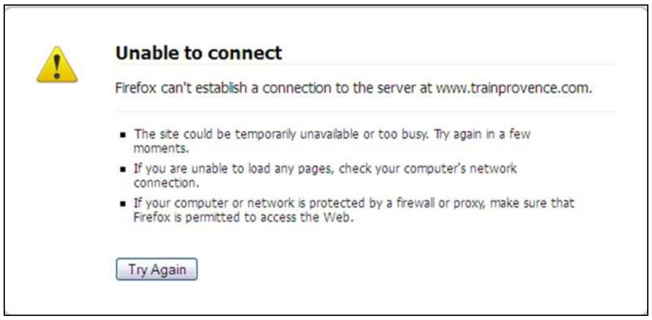
- The connection to RSS (Really Simple Syndication) feed aggregators (Google news, CNBC latest quotes) does not show any sign of break. Actually, the pages keep refreshing themselves, permanently sending new HTTP requests, each of them establishing a new TCP connection.
- Another application tested was a webcam server. In this case, the video flow is broken and cannot be recovered, as shown in Figure 1.



Connection closed

Figure 1: Disruption on a browser application

- The last application tested returns the timetable of a train journey. A first request is performed while being connected to the Wi-Fi access. In the case of test A, there is no impact, the second request being sent directly on a new connection. In the case of test B, the second request is sent while no network connection exists and returns an error message as shown in Figure 2. There is no recovery possible to this case, retrying the same request always brings the same reply. The only possible solution is to re-open a new web session in another window, and then navigate through the website again to get back to the same page. In the case of test C, the second request returns the new timetable with an additional delay of 3s.



Unable to connect

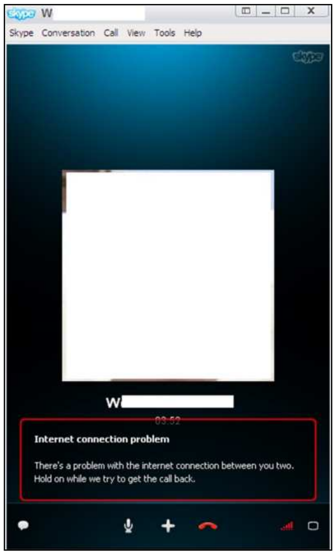
Firefox can't establish a connection to the server at www.trainprovence.com.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Figure 2: Disruption on a web service application

VoIP

Skype was used to test voice communications. For tests A and B, the application hangs out when disconnected, as shown in Figure 3 and restarts automatically, with a delay lower than 3s, when a new connection is found. In the case of test B, the application keeps trying to connect, whatever the duration of the disconnection, which consumes uselessly the battery of the terminal. In the case of test C, no impact can be observed. If the disconnection happens during a voice conversation, Skype re-dials the called number by itself to re-start the phone communication.



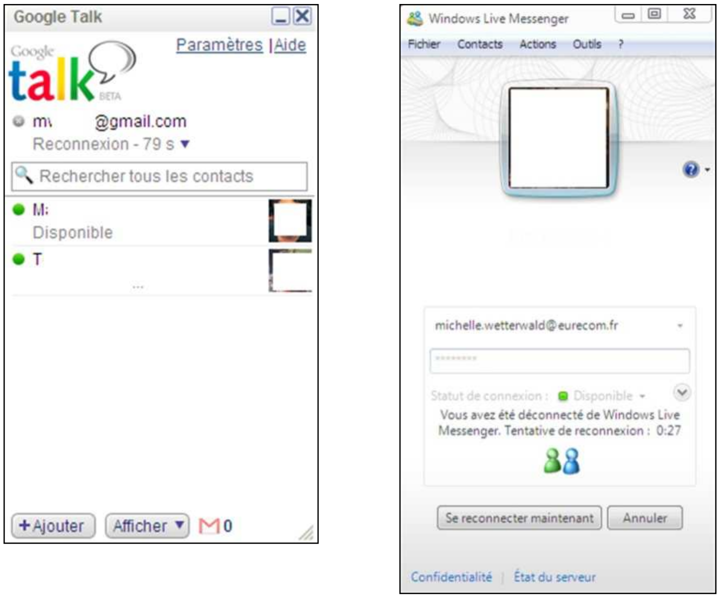
Internet connection problem

There's a problem with the internet connection between you two. Hold on while we try to get the call back.

Figure 3: Disruption on Skype application

Instant Messaging

Two different applications were tested for Instant Messaging (IM), showing an identical behaviour. For tests A and B, when disconnected, the application starts a timer (see Figure 4) with a value comprised between 20 and 30s. When the timer expires, the IM application tries to reconnect. If it fails, the timer is started again, with a higher value (from 40 to 90s). And so on, until the second technology appears. The timer is immediately reduced to a value of 3s, to confirm the new connection, and then the communication is re-established automatically. In the case of test C, there is a small break of 1-2s, after which the application is transferred to the new connection.



m\ @gmail.com
Reconnexion - 79 s ▾

● M:
Disponible

● T

michelle.wetterwald@eurecom.fr

Statut de connexion : Disponible ▾

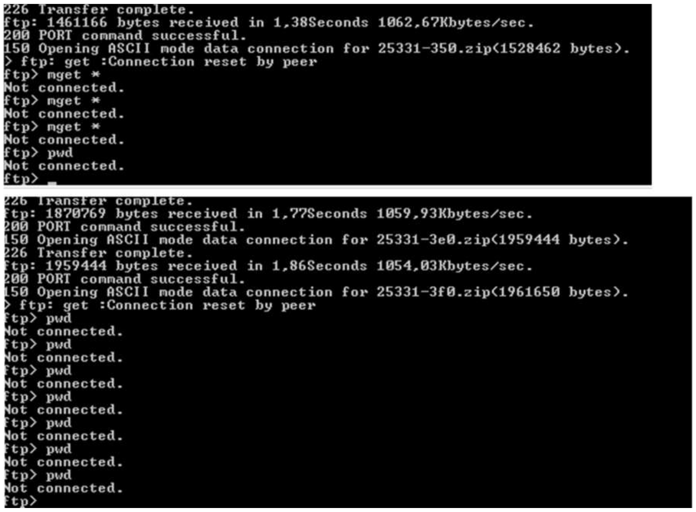
Vous avez été déconnecté de Windows Live Messenger. Tentative de reconnexion : 0:27

Figure 4: Disruption on Instant Messaging applications

FTP

Evaluating different file transfer applications gives a good idea about how the problem of disruption is addressed by the application developers. The tests were conducted with two applications (Command line FTP and FileZilla), while the third one (BitTorrent) was studied on paper only due to regulatory issues.

When using a standard FTP session at the console, it is broken and unable to recover whatever the test, as shown in Figure 5.

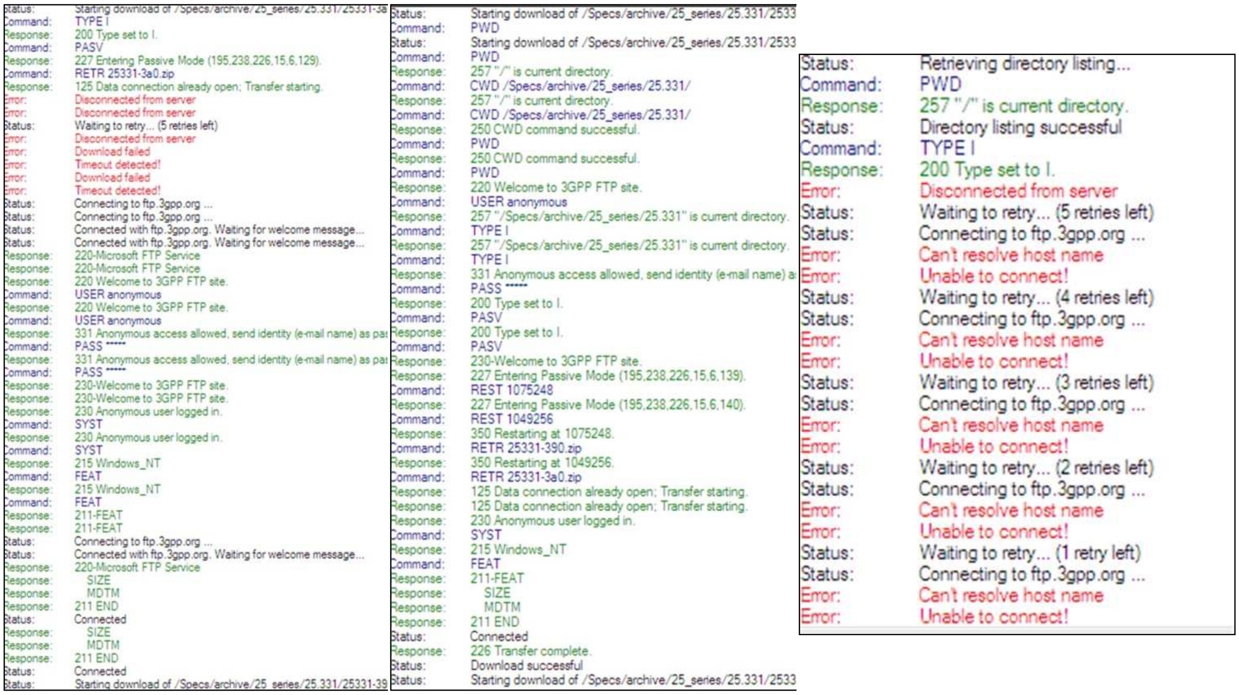


```
226 Transfer complete.
ftp: 1461166 bytes received in 1,38Seconds 1062,67Kbytes/sec.
200 PORT command successful.
150 Opening ASCII mode data connection for 25331-350.zip(1528462 bytes).
> ftp: get :Connection reset by peer
ftp> mget *
Not connected.
ftp> mget *
Not connected.
ftp> mget *
Not connected.
ftp> pwd
Not connected.
ftp> _
```

```
226 Transfer complete.
ftp: 1870769 bytes received in 1,77Seconds 1059,93Kbytes/sec.
200 PORT command successful.
150 Opening ASCII mode data connection for 25331-3e0.zip(1959444 bytes).
226 Transfer complete.
ftp: 1959444 bytes received in 1,86Seconds 1054,03Kbytes/sec.
200 PORT command successful.
150 Opening ASCII mode data connection for 25331-3f0.zip(1961650 bytes).
> ftp: get :Connection reset by peer
ftp> pwd
Not connected.
ftp> pwd
Not connected.
ftp> pwd
Not connected.
ftp> pwd
Not connected.
ftp> pwd
Not connected.
ftp> pwd
Not connected.
ftp> pwd
Not connected.
ftp>
```

Figure 5: Disruption on a standard FTP exchange

FileZilla is an FTP client application (it may also provide the server side when needed) offering a GUI to hide and launch the FTP commands. The observation shows that it stores the current FTP status: FTP server address, user credentials, current directory, last command or set of commands issued. When the connection is broken, it starts a small timer of 3s. When the timer expires, it is retried with a slightly increased value of 5s. This process is repeated 5 times, after which the connection is declared broken, an error is raised and only manual intervention can recover from the root of the file server tree. This happens with test B. In test A, the second access is actually connected before the 5th time-out. Using the stored information, FileZilla is able to restart its execution directly at the last completed action. In test C, there is a small break of 2s, after which the FTP client restarts where it was interrupted. These behaviours are pictured in Figure 6. The screenshot on the left shows a disconnection followed by a re-connection; the screenshot in the middle shows that the application kept a history of the last operations and is able to restart where it was disconnected; the screenshot on the right shows the expiration of the 5th time-out and the final error message.



```
tatus:      Starting download of /Specs/archive/25_series/25.331/25331-3a
ommand:     TYPE I
esponse:    200 Type set to I.
ommand:     PASV
esponse:    227 Entering Passive Mode (195,238,226,15,6,129).
ommand:     RETR 25331-3a0.zip
esponse:    125 Data connection already open; Transfer starting.
rror:       Disconnected from server
rror:       Disconnected from server
tatus:      Waiting to retry... (5 retries left)
rror:       Disconnected from server
rror:       Download failed
rror:       Timeout detected!
rror:       Download failed
rror:       Timeout detected!
tatus:      Connecting to ftp.3gpp.org ...
tatus:      Connecting to ftp.3gpp.org ...
tatus:      Connected with ftp.3gpp.org. Waiting for welcome message...
tatus:      Connected with ftp.3gpp.org. Waiting for welcome message...
esponse:    220-Microsoft FTP Service
esponse:    220-Microsoft FTP Service
esponse:    220 Welcome to 3GPP FTP site.
ommand:     USER anonymous
esponse:    220 Welcome to 3GPP FTP site.
ommand:     USER anonymous
esponse:    331 Anonymous access allowed, send identity (e-mail name) as pas
ommand:     PASS ******
esponse:    331 Anonymous access allowed, send identity (e-mail name) as pas
ommand:     PASS ******
esponse:    230-Welcome to 3GPP FTP site.
esponse:    230-Welcome to 3GPP FTP site.
esponse:    230 Anonymous user logged in.
ommand:     SYST
esponse:    230 Anonymous user logged in.
ommand:     SYST
esponse:    215 Windows_NT
ommand:     FEAT
esponse:    215 Windows_NT
ommand:     FEAT
esponse:    211-FEAT
esponse:    211-FEAT
tatus:      Connecting to ftp.3gpp.org ...
tatus:      Connected with ftp.3gpp.org. Waiting for welcome message...
esponse:    220-Microsoft FTP Service
esponse:       SIZE
esponse:       MDTM
esponse:    211 END
tatus:      Connected
esponse:       SIZE
esponse:       MDTM
esponse:    211 END
tatus:      Connected
tatus:      Starting download of /Specs/archive/25 series/25.331/25331-39
```

```
Status:     Starting download of /Specs/archive/25_series/25.331/2533
Command:    PWD
Status:     Starting download of /Specs/archive/25_series/25.331/2533
Command:    PWD
Response:   257 "/" is current directory.
Command:    CWD /Specs/archive/25_series/25.331/
Response:   257 "/" is current directory.
Command:    CWD /Specs/archive/25_series/25.331/
Response:   250 CWD command successful.
Command:    PWD
Response:   250 CWD command successful.
Command:    PWD
Response:   220 Welcome to 3GPP FTP site.
Command:    USER anonymous
Response:   257 "/Specs/archive/25_series/25.331" is current directory.
Command:    TYPE I
Response:   257 "/Specs/archive/25_series/25.331" is current directory.
Command:    TYPE I
Response:   331 Anonymous access allowed, send identity (e-mail name) a:
Command:    PASS ******
Response:   200 Type set to I.
Command:    PASV
Response:   200 Type set to I.
Command:    PASV
Response:   230-Welcome to 3GPP FTP site.
Response:   227 Entering Passive Mode (195,238,226,15,6,139).
Command:    REST 1075248
Response:   227 Entering Passive Mode (195,238,226,15,6,140).
Command:    REST 1049256
Response:   350 Restarting at 1075248.
Command:    RETR 25331-390.zip
Response:   350 Restarting at 1049256.
Command:    RETR 25331-3a0.zip
Response:   125 Data connection already open; Transfer starting.
Response:   125 Data connection already open; Transfer starting.
Response:   230 Anonymous user logged in.
Command:    SYST
Response:   215 Windows_NT
Command:    FEAT
Response:   211-FEAT
Response:      SIZE
Response:      MDTM
Response:   211 END
Status:     Connected
Response:   226 Transfer complete.
Status:     Download successful
Status:     Starting download of /Specs/archive/25_series/25.331/2533
```

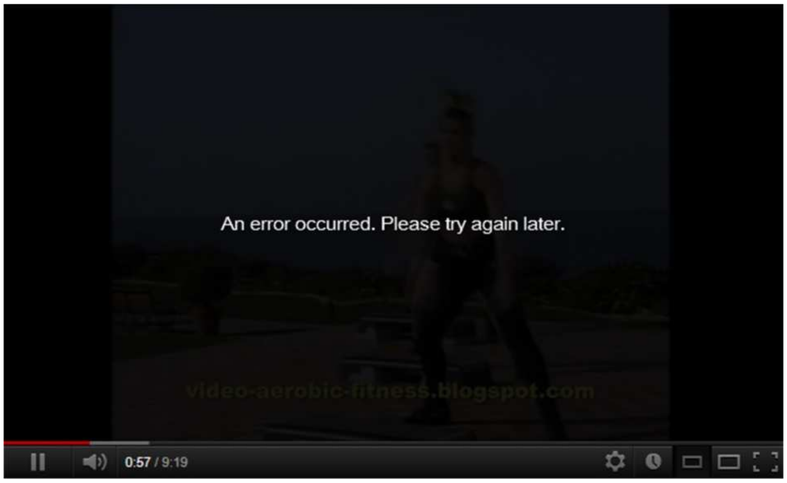
```
Status:     Retrieving directory listing...
Command:    PWD
Response:   257 "/" is current directory.
Status:     Directory listing successful
Command:    TYPE I
Response:   200 Type set to I.
Error:      Disconnected from server
Status:     Waiting to retry... (5 retries left)
Status:     Connecting to ftp.3gpp.org ...
Error:      Can't resolve host name
Error:      Unable to connect!
Status:     Waiting to retry... (4 retries left)
Status:     Connecting to ftp.3gpp.org ...
Error:      Can't resolve host name
Error:      Unable to connect!
Status:     Waiting to retry... (3 retries left)
Status:     Connecting to ftp.3gpp.org ...
Error:      Can't resolve host name
Error:      Unable to connect!
Status:     Waiting to retry... (2 retries left)
Status:     Connecting to ftp.3gpp.org ...
Error:      Can't resolve host name
Error:      Unable to connect!
Status:     Waiting to retry... (1 retry left)
Status:     Connecting to ftp.3gpp.org ...
Error:      Can't resolve host name
Error:      Unable to connect!
```

Figure 6: Disruption on a file transfer client application

BitTorrent brings an additional level of improvement in the handling of file transfer, also called Delay Tolerant Data Transfer. What is peculiar from the P2P technology is that the files are retrieved from other client applications (peers) which may get shutdown at any time. Each file that can be retrieved by the client is generally divided into blocks of 256k octets, each of them being further divided into 8 pieces. Each piece is identified in a descriptor file which contains also a hash to check the integrity of the received file before re-combination. The hash is tracked by the application which knows which pieces were already obtained, which ones are missing and where they reside at a certain point in time. Because of this P2P constraint, when reconnected, the client is able to start again downloading the missing pieces and no impact is observed.

Streaming

Streaming applications bring a certain level of resiliency in the handling of video download because they buffer the received frames until they are shown on the screen. Even though the video itself is transported on an UDP connection, the control runs on TCP. For every test, it has been observed that the video continues until the end of the buffer. Then the video stops and is unable to start again, even manually, on the new connection. The application must be restarted from its beginning. This is illustrated in Figure 7.



An error occurred. Please try again later.

Figure 7: Disruption on a streaming application

Secured Application

The test was performed using an SSH client. Whenever the application is disconnected, it stops and cannot start over on another connection, whatever the test, and even in the test C when both interfaces are available. The connection has to be re-established manually, typing again the user credentials. This is a case where the break of the TCP session is exploited by the application to guarantee the security of its user.

Internet Cloud Applications

The test was performed using the Dropbox application to access a non-secured storage in the cloud. Dropbox behaves exactly as an IM application, and even better, since there is no delay between the setup of the new connection and the green status of the application icon on the screen.

The test has also been conducted with another similar application which additionally includes a secured access to the cloud storage. For all the tests, the client application stops and is not able to restart the download after the disconnection. However, a new request for download can be successfully performed after the new connection is set up.

2.1.3 Tests Wrap-up and Conclusion

These tests have shown that whatever the application, when the access to the network is disconnected, the application connectivity is always broken. Some applications recover if the delay to establish the new connection is short enough, some even restart by themselves and are able to go back to where they were broken (e.g. IM, Skype, FileZilla or BitTorrent), at the cost of a higher level of consumption for the CPU (Central Processing Unit) time and battery. Most applications are completely stopped (streaming, FTP, SSH). It may even happen that, if it was not developed properly, the application enters in an undefined state.

Figure 8 shows a summary of the results obtained. Each application (coded with a letter to improve the figure readability) is shown in each of the test. As an example, Filezilla is coded as D2 and is shown in the first group (immediate recovery) for test A, in the last group (no recovery) for test B and in the second group (recovery after 3 seconds) in test C.

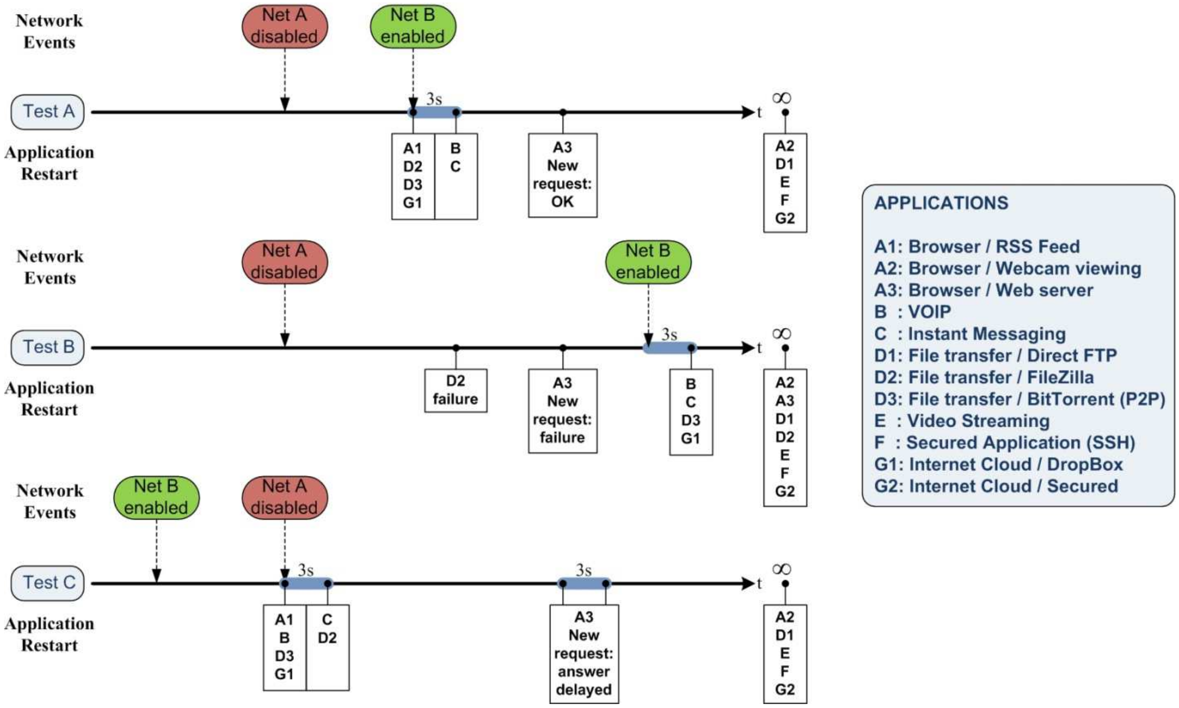


Figure 8: Application Recovery Timeline

Currently, most of the existing Internet applications are running over TCP as transport protocol with possibly other protocols between the TCP socket and the application (e.g., Secure Sockets Layer or SSL, XMPP…). TCP is a connection-oriented protocol. When the connection is broken, the packets in the network are lost, timers in the protocol elapse and the session is closed. TCP is not able to recover the connection break by itself. It has to be restarted. In the current terminals, it is the responsibility of the application to take care of the connectivity and packets recovery in case of a connection break. This is of utmost importance when a mobile terminal moves between two independent networks and cannot be assisted by a mobility protocol. The next sections will introduce some existing mechanisms which attempt to solve, at least partly, this problem.

2.2 Heterogeneous Networks and Media Independent Services

End-user mobile terminals are becoming more and more complex. Part of this complexity is caused by the objective to access heterogeneous networks and include the devices necessary to support more than one access technology. Operating multimode devices in heterogeneous networks can become very complex if each access technology has to be addressed directly and separately by the networking entities. As applications and upper layers of the control plane should remain independent of this evolution, there is an increasing need to design abstraction modules, providing a generic interface to the upper layers and taking care of the complexity of operating the various interfaces in the terminal. Depending on the application requirements, intelligent interface selection is needed while the system is in operation. At the same time the specificities of the access technologies must be hidden to the applications. Moreover, a decoupling approach must be undertaken in order to ease the process of incorporating future evolutions of access technologies. Seamless handover between heterogeneous access points should be implemented using uniform mechanisms, with an abstract interface specified to allow informing the upper layers of the capabilities of the underlying technology as well as configuring these capabilities. Technology, operation and policies can be managed specifically for each type of network below that interface.

To solve this issue, the IEEE, through its 802.21 work group [22], has developed a standard that allows a MT to seamlessly roam across different types of 802 network access technologies, such as 802.11 (Wireless Local Area Network, or WLAN) and 802.16 (Worldwide Interoperability for Microwave Access, or WiMAX). In addition, the Media Independent Handover (MIH) Services [21] enable the mobility to and from cellular systems based on 3GPP and 3GPP2 specifications. The main reason for introducing this group was mostly that the 802.x set of standards permits handover only to the same network types and behind the same router. The standard was thus designed to facilitate operations related to mobility and provide a technology-independent interface just below the network layer of the ISO/OSI stack, above the technology dependent data link and physical layers. This interface is not present only at the MT, but is also deployed for signalling between the MT and the different network entities involved in a handover scenario (AR – or Access Routers -, AP - or Access Points - and BS – or Base Stations).

The 802.21 proposes three different Media Independent Services [23] 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 measurements and dynamic changes in the different links. The Command Service (MICS) allows the upper layer management entities to control the behaviour of the different links. The Information Service (MIIS) is responsible for distributing to the MTs the technology-independent, topology-related information and policies from a repository located in the network. The MIIS allows the terminals to choose the optimal network, with the maximum amount of awareness of their wireless neighbouring environment.

A central component, the Media Independent Handover Function (MIHF), pictured in Figure 9, provides the other functions and modules running in the mobile with the MIH services. A cross-layer architecture is defined where the MIHF acts as a relay between *(i)* the media-specific Link layer entities connected by the MIH_LINK_SAP (Service Access Point) and *(ii)* the media-agnostic upper layer entities, or MIH-Users, connected over the MIH_SAP and responsible for handling the mobility protocols operation, adaptive applications or service logic (e.g. IMS, IP Multimedia Subsystem). A similar approach is present in the network entities, where the MIHF acts as an intermediary, abstracting the technology specific functions, and making them available over a technology-independent interface. This kind of deployment makes it easy for MTs to select the best Point of Attachment (PoA) since the MIHFs at the MIH_LINK_SAP take care of "translating" the messages for each technology.

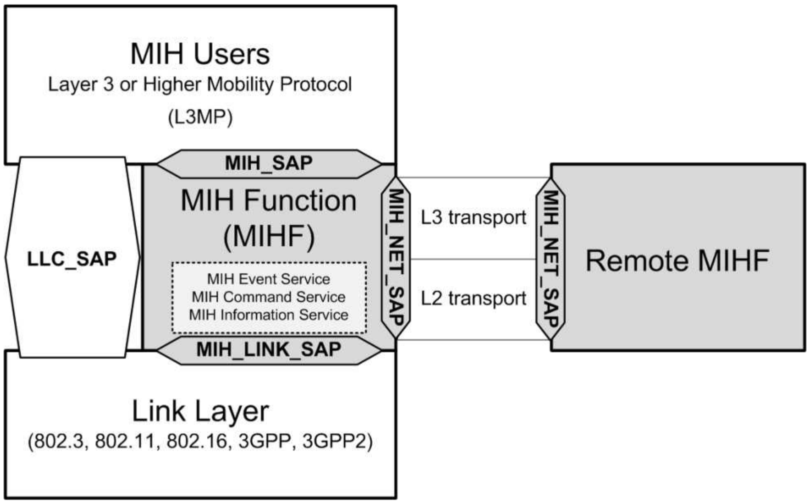


Figure 9: Reference Model for Media Independent Handover

The MIHF is also responsible of handling the MIH protocol that runs between the different network nodes to synchronize the MIH operations. This protocol provides rules for peer communications between the MIHF modules located in these nodes. It covers the MIHF capability discovery, in order to learn the information related to a remote node (supported events, commands, information types) complemented by the messages necessary to synchronise the preparation, execution and completion of the handover operation. The protocol operates through the MIH_NET_SAP, using either Layer2 or Layer3 transport, according to the access network.

In summary, as described in [24], the design of the 802.21 standard introduces three types of new elements: *i)* a framework which enables seamless handover and service continuity by providing mechanisms to collect and distribute all the information necessary to attach a new PoA, whatever its access technology; *ii)* new functions designed to facilitate the handover by exchanging information between the existing entities in the protocol stack in an abstract manner; *iii)* several Service Access Points interconnected by a central entity, the MIHF.

Currently, this standard is being amended to support an extended set of services, such as security (802.21a), single radio handover (802.21c) or group management solutions (802.21d). Additional technologies, such as downlink-only technologies (802.21b), mostly for broadcast, are considered. More recently, a proposal has been submitted to revise the general IEEE 802.1 architecture [25], for management, security and media independent handover by including a new Media Independent Control Function (MICF), as a parallel control plane entity in the IEEE 802.x reference model. It provides control functions for different MAC and PHY sub-layers. MIHF is one of them, as well as the control functions for the IEEE 802.19 (coexistence between unlicensed wireless networks) or 802.22 (Wireless

Regional Area Networks, WRAN, using cognitive radio in TV whitespaces). The design of this framework is particularly suited to control the operation of the upcoming multimode terminals. However, it should be noted that the standard, once considered for 3GPP cellular systems, has been pushed aside to the profit of the ANDSF (see Section 2.4). According to a network operator representative met during a Future Internet workshop, 3GPP considered mostly MIIS and preferred ANDSF. Other services, such as services located in the MT, could be used without impacting the network operators while bringing improvements to the handling of heterogeneity. Another important consideration is that the scope of IEEE 802.21 covers handover detection, initiation and preparation only. It does not cover handover execution or replace the existing mobility management functions and protocols (Mobile IP, SIP... see Section 2.3).

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 several strong limitations. It currently only enables handovers, is restricted to control plane operations and addresses exclusively network interfaces. It offers the possibility to be developed to support an extended set of services and devices in the terminal and operate on the data traffic as well, transferring the packets towards one of the available network interfaces. This extension will be a main axis for the design of the target solution.

2.3 Handling Mobility

The past ten years have seen the convergence of mobile communications with data communications, and more specifically, the Internet. Even the way mobility management is defined has strongly evolved. It initially started with nomadism where laptop users where roaming between access points but did not change their location after being connected. DHCP (Dynamic Host Configuration Protocol) was a good solution to provide them with connectivity anywhere. Then devices became smaller, with the arrival of connected PDAs (Personal Digital Assistants) and more recently of the smart phones, which imposed strong requirements for always-on connectivity to the Internet. The latest devices are now expected to be usable while walking on the streets, transported in road vehicles or even in fast speed trains. However, when the device moves out of its original routing area, it cannot maintain its IP address and the session is broken. 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, 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. Two types of handovers are defined in heterogeneous networks. A vertical handover is executed when the MT changes the type of technology it uses to access the network. Only multimode terminals with at least two network interface devices can execute vertical handovers. A horizontal handover is performed between two PoAs using the same technology.

The Session Initiation Protocol (SIP) can be used as a signalling protocol for Session and Application layer mobility service [26], independently of the access network technology. Using SIP, a session can be transferred between different terminals or access domains, thus enabling personal mobility. The SIP registration procedure is the key feature to provide service mobility and inform the SIP server of the location of the current session, together with its allocated IP address. [27] proposes a scheme where SIP, integrated with 802.21, is used on one hand as a signalling protocol for pro-active registration on the new visited network and on the other hand as a carrier for the MIIS signalling. The pro-active registration, performed as soon as an MIES event signal of network getting lost is raised, allows eliminating the authentication delay from the handover delay count, but requires some adaptation in the SIP servers. Another issue with SIP mobility is that it does not bear when the handover delay is too long. Other solutions were designed at transport layer level by altering the TCP protocol or introducing new protocols such as SCTP (Stream Control Transmission Protocol) dynamic address reconfiguration.

For terminal mobility at Network layer level, IETF groups introduce mechanisms focusing on handoff and routing aspects of the wireless networks, the MIPv4 and MIPv6 (Mobile IP for IPv4 and IPv6) protocols. They provide a centralized macro-mobility solution where the MT, identified by a Home Address (HoA), binds using a temporary Care of Address (CoA) to an entity called the Home Agent (HA), located in the MT's home network [28]. The HA receives the packets in place of the MT and forwards them to its new location. When the MT moves, the new CoA is provided to the HA which just needs to update its binding cache, operation that remains transparent to the Correspondent Node (CN). This mechanism requires the availability of a HA entity in the home network of one of the technologies, administered by the operator of that network or by another recognized organization. Many drawbacks have been found to this solution: packets lost during the handover, HA being a single point of failure, overhead due to the necessary IP tunnels between the HA and the MT, inefficient triangle routing, low performance in case of local mobility, security holes, etc. A large set of fixes and enhancements towards a seamless handover has been designed [S-VHO] [29], introducing the improvements described below, i.e. Fast MIP, HMIP, HIP, PMIP, DMM or flow mobility.

The Fast Mobile IP protocol (FMIP) [30] activates a tunnel between the previous AR and the new AR before the handover execution. This tunnel holds for the time of the handover, packets are saved at the new AR and delivered when the MT has completed its attachment to the new AR. It reduces the packet loss and accelerates the handover, which is essential for real-time traffic. Hierarchical MIP (HMIPv6) [31] is designed to improve the handoff speed by separating local or micro mobility from global mobility. It introduces a local entity, the MAP (Mobility Anchor Point) to manage the local handovers, while global mobility between different MAP domains is handled by MIPv6. The Host Identity Protocol (HIP) [32] is a mobility solution operating at layer 3.5 (between Network and Transport layers), based on host identities instead of IP addresses, both of them bound dynamically according to the device movements. It has been noted that HIP introduces some security issues and requires a change of the DNS (Domain Name System) servers in the network. Proxy MIP (PMIP) enables mobility management without any change or impact on the MT protocol stack. Its functionality is fully located in the network, which is responsible for handling the mobility signalling on behalf of the mobile. However, in the case of a multimode terminal, some mechanisms are required in the mobile to maintain the same IP address across the various interfaces and the mobility anchors must be federated. Flow mobility is yet another solution. It does not work at per-node granularity (i.e. all traffic is mobility-enabled), but on a per-application (flow) basis. It provides mobility only to certain sensitive applications and can move different flows to different interfaces. The Distributed Mobility Management (DMM) concept [33] is currently under discussion at the IETF. It flattens the mobility architecture by placing the anchors closer to the MT, even directly at the AR level. A key property of most of these mechanisms is that they operate in the network part of the system, removing the handover control from the MT with the objective to keep it as less affected as possible. The counterpart is that they require major changes in the existing network entities or the introduction of new control entities which have to be maintained by a well-identified organization and are thus difficult to deploy.

In cellular systems, mobility is handled by 3GPP proprietary protocols and procedures. These protocols enable mobility between the different RATs (Radio Access Technologies) supported, i.e. 2G, 3G or LTE (Long Term Evolution) [34] and with non-3GPP accesses, WiMAX or CDMA2000 [35]. Inside the LTE system, local mobility is “X2-based”, i.e. handled directly between PoAs (eNodeB, or cellular base station), the MT remaining under control of the same Mobility Management Entity (MME). If the MME has to be changed, the mobility is “S1-based” and implies a change of Serving Gateway (S-GW) or even Public Data Network Gateway (PDN-GW). The IP address of the mobile is allocated when the default bearer is activated when the terminal attaches to the mobile network. Even when IPv6 stateless auto-configuration is used, the IPv6 and interface identifier used to build this address are provided by network entities. The same address is used for the subsequent bearers and kept throughout the same PDN connection. In case a new PDN connection has to be established due to the mobility procedure, all active application sessions are broken and have to be re-started or recovered. However, this case is considered as very unusual by mobile operators, since a PDN-GW usually covers a very large geographical area. Connections external to the mobile operator, e.g., from a campus network, are not included in the cellular mobility mechanisms.

A different, but interesting concept, nonetheless, has been described in [36]. The authors introduce the notion of Personal Address (PA), which is associated to the user rather than the device, and maintained for the whole duration of the session. It isolates completely the session from the terminal location and allows a full transparent mobility from all points of view: device, session, and even application. The CN involved in the session sees the same address over the whole duration of the session. The associated architecture is pictured in Figure 10. Since this address does not reflect the location of the current PoA, a Delivery function is introduced in the network to locate the user and forward the packets to his current location. A Migration function located in the MT maintains the Delivery function updated with the current location of the user and his session. The Adaptation function, also located in the network, adapts the application parameters in case of a session handover between different terminals.

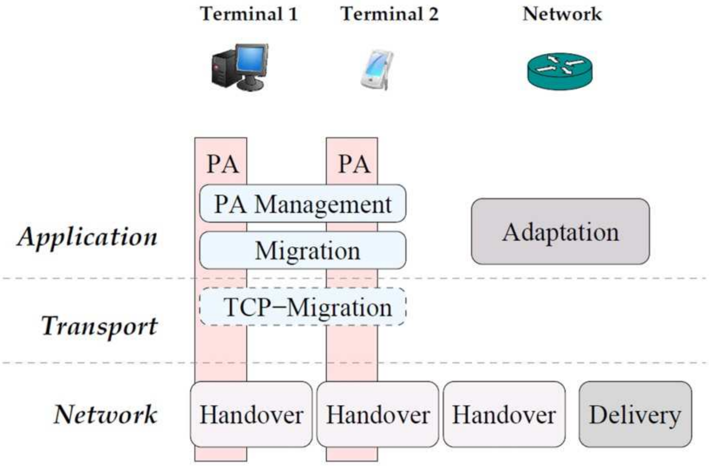


Figure 10: Personal Address mobility framework (from [36])

Taking this concept as a preliminary base, it is now possible to separate completely the applications from the network connection and the mobility processes. The terminal can control its own mobility, hiding it to the application and does not rely on any specific networking protocol. The mobility processes can perform handover, assisted by tools such as security mechanisms for example, to ensure that the session is never broken for any reason, thus enabling Personal Networks to be set-up and managed. In the same type of idea, [37] proposes to handle mobility with a Virtual Network Address Translation (VNAT) architecture, based on the fact that IP and the transport protocols above it (UDP, TCP) were designed for stationary devices. VNAT virtualizes the end-to-end connections by using virtualized identifiers at application level, rather than the usual IP address associated with a port number. These identifiers are mapped into appropriate physical identifiers according to the terminal connectivity in a specific component located at Network layer. A third component, called "helper" and located at the network layer of the server is designed to hide network disruptions to the applications by suppressing the keep-alive timer of the TCP connection or temporarily suspending the client-related process. When the connection resumes, the action taken is reversed. [38] proposes a session layer Mobility solution made of two components, the CAS (Connection Abstraction System) which handles the mobility at Session layer (Layer 5) and the AMS (Address Management System) which uncouples the application from the rest of the protocol stack by hiding and adapting addresses and lower protocols used according to the existing environment under generic addresses provided by the system.

Another interesting technique is based on Delay Tolerant Networks which allow the mobile nodes to survive connectivity disruptions. Actually, the DTN abbreviation is often used to name Disruption-Tolerant Networking as well. This technique was designed for satellite and outer-space communications, where wireless networks are power-limited, mutually incompatible, support long and variable delays and may face periods with high error rates or link disconnections. The impact of an intermittent connectivity is overcome by using store-and-forward message switching [39] [40]. Whole or pieces of a specific message are moved between persistent storage nodes (called DTN nodes), which store the message for long periods of time until they are able to forward it to the next DTN node. In the forwarding process, DTN nodes thus replace the routers, but actually operate like gateways above the transport protocol. Indeed, a DTN node is rather the opposite of a standard IP router which keeps the packets for a few milliseconds only, the time required to lookup in its tables for the route towards the next hop.

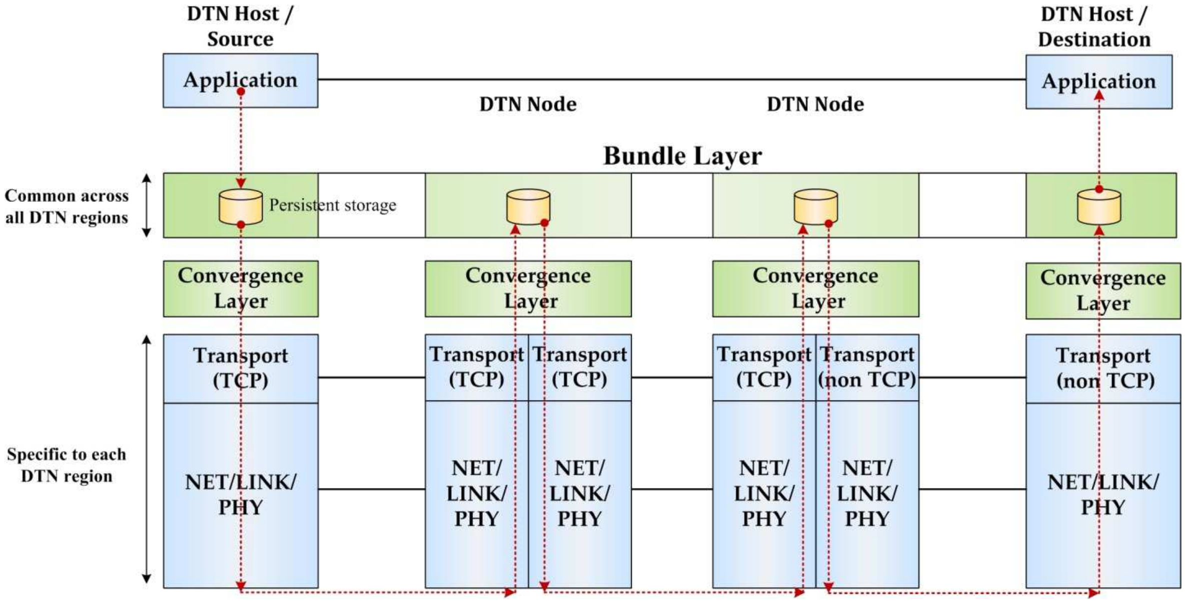


Figure 11: Store-and-forward message switching in DTN nodes (adapted from [39])

The store-and-forward functionality is provided in the DTN nodes by an end-to-end message-oriented overlay, called the “Bundle layer”, which is inserted between the application and the transport layer of the nodes supporting this technique, as shown in Figure 11. The bundle layer is actually layer-agnostic and focuses on forwarding virtual messages (the bundles, or pieces of the real message), rather than packets. Because the TCP protocol provides an end-to-end connectivity (source-to-destination), it is used as transport protocol only to insure the reliability of the communication on a network segment. DTN nodes terminate transport protocols at the Bundle layer, which makes the DTN architecture tolerant to delay and connectivity problems. In cases where TCP cannot be used in a DTN region, a Convergence layer adapts the Bundle layer to the available transport protocol. DTN techniques were designed for outer space communications, but they are also used for opportunistic mobile ad-hoc networks such as ITS (see Section 2.6), public transportation where the vehicle acts as a store-and-forward node to the MTs inside, military ad-hoc networks in hostile environment or Wireless Sensor Networks (WSN). They can cope with long disruption delays, which are oversized compared to a handover delay, but still can be simplified to optimize the target solution.

In [41], a Disruption Tolerant Mobility Architecture (DTMA) is proposed, adapted from the DTN architecture, to support the MT mobility, targeting both vertical and horizontal handovers. The DTMA introduces two new entities to shield the connectivity disruptions from the applications: a local proxy in the terminal and a proxy gateway in the network. SIP is used for the connection and application initiation signalling. The Convergence layer of the DTN architecture is capable of adapting the Bundle layer to TCP or UDP protocols. The communication between the application and the CN is split into two parts. The data connection from the user application terminates at the local proxy, and thus is not affected by a change of address or loss of connectivity. Next, the local proxy initiates a TCP or UDP socket connection towards the CN through the Internet. If the mobile is disconnected from the network, the proxy gateway located in the network receives the data on its behalf, stores them and transfers them only when a new connection is established. The proxy gateway and the mobile appear to the CN as one single DTN node, accessible via two paths of different costs.

In summary, many mechanisms have been designed which handle mobility inside a single domain or two cross-related domains. All of them introduce new functional entities or new protocols in the network. Some of them require tunnels which reduce the traffic efficiency and thus the bandwidth left to the application itself. Moreover, they often rely on specific entities owned by the network operator and do not perform well when moving to or from a network not supported by that same operator. At the end, session continuity relies on the capacity of the application to recover by itself from disruptions caused by non-covered mobility cases, as was demonstrated in Section 2.1. Similarly to DTN architecture, our objective is to bring this continuity under the control of new generic facilities above the network and transport protocols and keep it independent of handover mechanisms existing at IP layer or above, while complying with the same requirement of surviving a change of the IP address.

2.4 Access Network Selection in Multimode Terminals

A major challenge of handling connectivity in heterogeneous networks with multimode terminals is to obtain a successful network access selection. This procedure must be executed when the MT starts the attachment to a network or when the current network performance is not satisfactory anymore and a decision to execute a vertical handover is made. It implies the choice of a new communication context to support the upcoming or running sessions and must be smarter than those based on simple signal strength only, which may lead to connect to the wrong one of the available access networks. A research in literature on heterogeneous wireless network selection returns a plethora of studies and surveys for vertical handover management and optimization algorithms. A vast majority of the results obtained take the terminal point of view, optimizing network access selection in conjunction with one of the mobility mechanisms described in Section 2.3. However, from a more general perspective, three cases can be envisioned *(i)* the mobile makes the decision on its own, based on its vision of the wireless neighbourhood, *(ii)* the mobile is assisted in this selection by network mechanisms such as MIIS or ANDSF (see below), *(iii)* the access selection is performed by the network, which, for example, allows the operator to balance its traffic load.

Figure 12 shows a mobile terminal at the border of two access networks, LTE and Wi-Fi, which has to decide which path is more efficient to reach the application server located across the Internet. This decision procedure, whose goal is to select a new or an additional wireless access, can be divided into four steps: *(i)* input collection, *(ii)* selection algorithm and decision making, *(iii)* output parameters validation, and optionally, *(iv)* user confirmation.

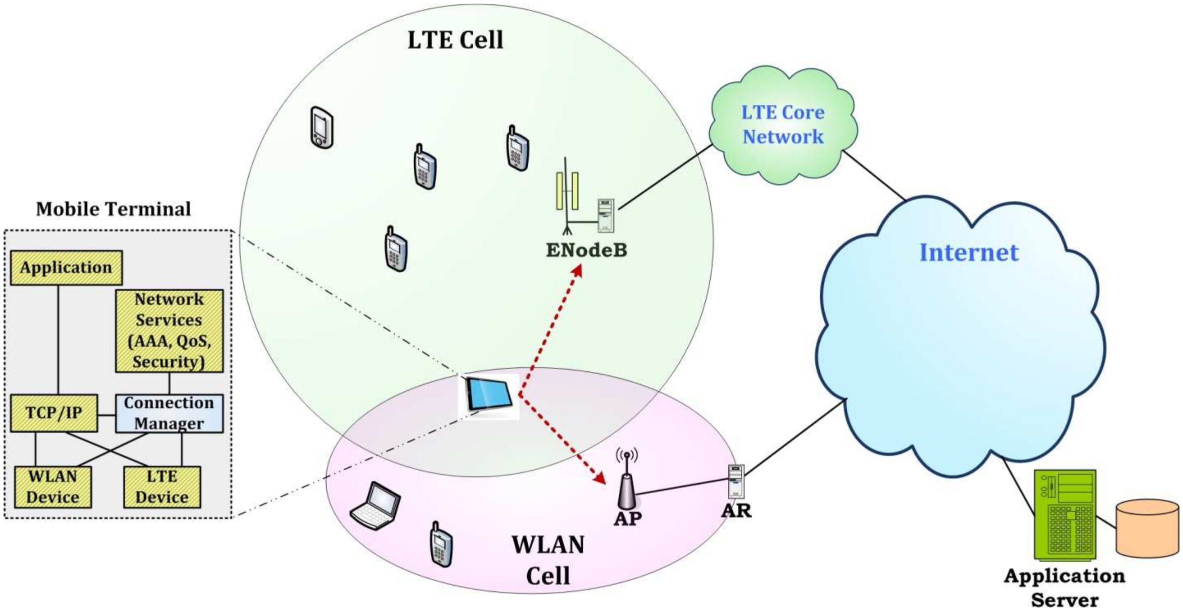


Figure 12: Example network setup involving access network selection

[42] provides a survey of vertical decision algorithms. The first step consists in the collection of the appropriate information to enable the decision process, according to a pre-defined list of criteria or attributes. In this survey, the attributes considered are all available from local resources, giving more importance to the user perspective: received signal strength, network connection time, available bandwidth, power consumption, monetary cost, security, and user's preferences. [43] proposes also a user-centric solution where users select the access network which best meets their data transfer performance for non-real time applications. The criteria used are very similar: terminal capability, data transfer requirements and communication rates. In [44], the decision is based on the same set of parameters, including as well the type of application running and its QoS (Quality of Service) requirements. [45] has the objective to optimize the performance of the system. Its Convergence Manager selects an interface for a generic file download service. The decision is made locally in the mobile to avoid any impact on the network, hiding from the application or the user the complexity of spreading traffic over different access networks. The selection is performed using attributes linked to the user context and mobility, such as access network coverage, current availability status, cost of connection, required QoS and load balancing. [46] targets a seamless mobility management approach under the full control of the terminal. The network selection only uses information available locally: access network identity, cost, battery lifetime, handover frequency and signal quality: SNR (Signal to Noise Ratio) or SINR (Signal to Interference plus Noise Ratio). [47] proposes a system made of two components, a Connection Manager (CMgr) and a Virtual Connectivity entity. The Connection Manager, by using novel sensing techniques at MAC (Media Access Control) and PHY layers, can obtain an accurate network condition to decide on the necessity of a handover.

Other studies enlarge the acquired knowledge of the context by retrieving information remotely from the network. [48] addresses all-IP environments to ensure service delivery at a certain location. The authors differentiate two categories of parameters, *(i)* those which do not change often and can easily be provisioned from the network, such as operator name, available services, or coverage area, and *(ii)* those that are QoS-related and more dynamic. In [49], a scheme enables the distribution of the flows between multiple interfaces. Each flow is associated to the network maximizing a utility function. The terminal runs IEEE 802.21, which provides information from the interface and from the network. The decision algorithm retrieves parameters from the network, in addition to the interface characteristics, QoS, throughput, power consumption, cost, delay, BER (Block Error Rate) and jitter. The specification in [50] proposes a Connection Manager to choose one or multiple paths suitable to a mobile protocol like MIP. It takes into account network interface information such as signal strength monitoring or other rules based on policies obtained from ANDSF, a GUI or application requirements. All the policies are stored in its database according to a pre-defined common format. Databases gathering policies and network information have been standardized by organizations such as the IEEE or the 3GPP. In the IEEE, the 802.21 MIIS is particularly suited for this operation. It indicates the different access networks available, the neighbour maps and network services. An equivalent system has been standardized for the mobile operator networks, the ANDSF [35]. This mechanism contains the data management and control functionality necessary for providing network discovery and selection assistance data to the MT as per operators' policy. It is able to initiate information transfer to the MT based on network triggers, and respond to requests from the MT. It provides inter-system mobility policies, and for the neighbouring networks, the access technology type and the access network identifier, together with the necessary operating parameters. Its objective is to assist the MT when performing vertical handovers.

Concerned with the MT operation, the Open Mobile Alliance (OMA) recently released some standards (still at candidate level) [51] [52]. They propose a common API for synchronous and asynchronous interfaces to applications and user interfaces in the terminal, covering security, cellular (2G, 3G, CDMA200) and WLAN connections, statistics handling, information status handling, GNSS (Global Navigation Satellite System) handling, power management... Figure 13 summarizes this OpenCMAPI architecture. It shows the functional components which provide generic services to the mobile proprietary applications. The objective is to simplify the design and development of applications and higher layer functionalities across all types of devices, mobile broadband devices, laptops, wireless routers, smartphones, tablets, cloud devices, requiring access to the mobile Internet and other technologies. However, the lower interfaces with the different hardware devices remain dependent on their implementation specificities. Moreover, services and functions are not inter-correlated and are executed as separate entities.

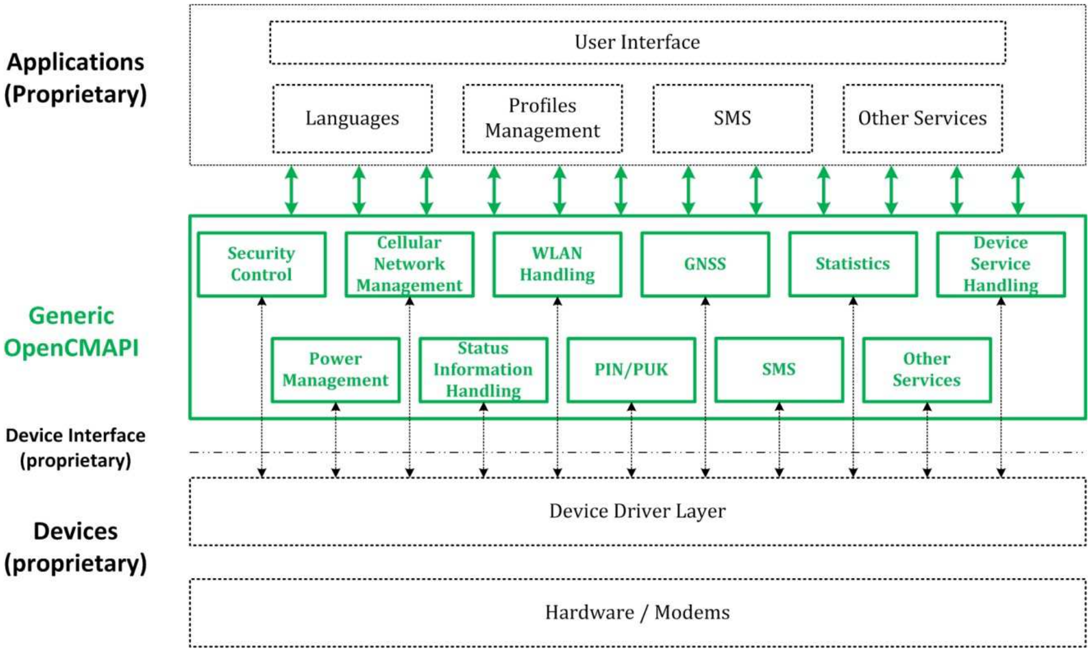


Figure 13: OpenCMAPI architecture for the Connection Manager (from [52])

At the IETF, a recent working group has also taken the initiative to solve issues related to multimode terminals. The purpose of the MIF (Multiple Interfaces) group [53] is to solve and standardize the amendments to existing mechanisms required at network level to support the attachment to multiple provisioning domains. At the time of completing this study, the group has released two RFCs, one stating the issues to be solved (RFC6418 [54]) and a second one surveying the behaviour of a subset of existing commercial operating systems and terminals (RFC6419 [55]).

The RFC6418 identifies the issues encountered at Network layer level by multimode terminals. It defines the scope of the working group, eliminating the support of lower layer interfaces and the algorithms necessary to select the best network interface. It provides a characterization of an MIF node, defined as a node capable to connect to multiple provisioning domains, either through IPv4 or IPv6 while using several physical or virtual interfaces. It can handle simultaneously several IP addresses and receives its configuration policies from the network. It uses MIPv6 to transfer the session when required. Even though not completely fixed yet, it appears that the group does not plan to address the access network selection. They refer to other standards for that task. In any case, the selection is expected to be made based on Link layer information, complemented with policies obtained from the network operator, user preferences, IP connectivity checking and application QoS requirements. They introduce as well the same notion of Connection Manager as mentioned in the previous paragraphs. Path failures from TCP breakdown or MIP failover are expected to be recovered by the application. This is out of scope of the planned work because they focus on networking mechanisms. Several issues related to the MIF nodes are described, including DNS resolution among the multiple domains, the choice between the different routes available on the active interfaces, potential conflicts between the policies retrieved from the different networks, network authentication and source address selection.

The RFC6419 describes the specific behaviour of several existing operating systems from two groups: mobile handsets and personal computers. It presents the current approaches for connection management, configuration of the connections according to the selection of the address or of the first hop by the applications. It shows that these are often conceptually different from one product to another.

The group's current activities include the design of an extension to existing DNS server selection in the MT, in order to be able to use multiple namespaces. Another item aims at defining an API, the MIF API, which allows the application to choose the most convenient network interface according to the addressing and DNS configuration. User preferences are used to filter out unsuitable interfaces. Traffic offloading from cellular networks or DHCPv6 route options are additional topics under study. A proposal has been made to integrate the User MIH_SAP in the MIF API and make the link with the CM. In summary, this group is tackling the problems to be solved to obtain a successful handling of multiple interfaces by a node at Network layer level. Indeed, the access network selection algorithms, the mechanism used to interact with the network interfaces, or the break of the connection when mobility mechanisms are not operational, remain out of scope of this activity.

As stated before, the trend at network operators' standardization activities is to avoid any impact on the mobile terminals or user interactions. [28] describes handover management solutions in heterogeneous networks. Interworking architectures are classified in loose and tight coupling solutions. In loose coupling architectures, the heterogeneous networks are considered as peers and MIP is the basic mechanism for inter-system mobility. In this category, policy-based solutions operate using two main components: a context repository which collects information from the various parts of the network, including from the terminal, and an adaptability manager, enforcing network policies. Advanced context management permits to improve the handover management. All these solutions are based on the same concept of collecting a set of attributes in order to make the network selection: network QoS parameters, neighbour maps, terminal location, terminal capabilities and user perceived QoS requirements.

In tight coupling architectures, a WLAN is directly attached to the cellular network. [56] and [57] describe two mechanisms used in cellular systems to take advantage of Wi-Fi hotspots when the user is equipped with a multimode terminal. The benefit for the operators is a reduction of the traffic load inside its Core Network. The Local IP Access (LIPA) allows a direct access to local shared resources, printers or media servers through a HeNodeB (Home eNodeB) sub-system. The access has to be specifically requested by the terminal to its MME. The Selected IP Traffic Offload (SIPTO) is activated based on the user subscription and offloads specific parts of his data traffic towards the Internet, bypassing the operator network. The benefit for the user is often a larger available bandwidth. [58] describes an inter-domain "Overlay Network Topology", spanning several core domains and proposing services across these domains operating on top of traditional networks. It couples Content Aware Networks (CAN) with Network Aware Applications (NAA). The user environment is stored in an exhaustive User Profile, gathering all the information necessary for a service to be efficiently deployed and accessible, with parameters related to the terminal characteristics, the end user preferences, and his access network characteristics, together with the objective and subjective QoS metrics processed. These parameters are reported by the terminal to a Home-Box, which provide the content delivery, in order to realize context-aware functionalities

In the second step, when all the required parameters and inputs have been collected, a selection algorithm is executed that combines these parameters or applies policies to make a decision. Algorithms range from simple comparisons where the best signal quality is chosen, to more complex ones which smartly combine the additional parameters from the end user, the application or the network context. The authors in [46] define their algorithm as an “if then else” analysis of the information collected before the handover. In [43], the decision algorithm is based on predicted rates and delay of data transfer. [50] and [45] match the decision with the policies retrieved or set-up during the first step. [42] provides a survey of classical decision strategies in Fourth Generation networks, classifying them based on the handover decision criteria: RSS (Received Signal Strength) based algorithms, bandwidth based algorithms, cost function based algorithms, and combination algorithms. The study shows that the preferred input is usually the RSS, sometimes combined with bandwidth information. Cost functions are more complex and combined algorithms the most reliable, but at the cost of larger handover delays.

[59] evaluates and classifies the different existing vertical handover decision strategies. The result demonstrates that advanced evaluation functions and optimized architectures are needed to perform better handover decision making, for user satisfaction as well as for the efficient use of network resources. The strategies analysed include user-centric strategies (taking into account user preferences in terms of cost and QoS) or strategies resolving a Multi-Attribute Decision Making (MADM) problem. The paper surveys well-known methods such as SAW (Simple Additive Weighting), TOPSIS (Technique for Order Preference by Similarity to Ideal Solution), WP (Weighted Products) or AHP (Analytic Hierarchy Process). Fuzzy logic and neural network-based strategies allow dealing with imprecise data in the set of input parameters and can be combined with MADM methods to develop enhanced algorithms. [44] defines a method based on a Markov Decision Process (MDP), using a link reward associated with the QoS achieved by the mobile connection and evaluated against the cost of handover signalling. He compares it to more classical methods. The results show better handover performance, but the converging time of the algorithm is of the order of magnitude of minutes, which would pose a problem on a real mobile device . [60] extends this solution and introduces a Q-Learning approach based on the history of the MT and a continuous interaction with its environment to maximize the user’s QoE. Compared with other approaches, it optimizes the decision, reducing the number of unnecessary handoffs, at the cost of lengthy learning periods. [49] introduces an innovative algorithm, the DiA (Distance to ideal Alternative), which ranks the set of possible choices by comparison to a Positive Ideal Alternative. It improves the result of SAW or TOPSIS thanks to an additional analysis of the possible results. [61] describes an integrated strategy, combining several simpler methods which take into account tunnelling costs or mobility signalling.

In the third step, the result obtained as output of this algorithm is made of a set of parameters which are needed to execute the decision, and includes for example the technology type, the protocols to be used or the QoS class to be requested to optimize the system performance. This step also includes the checking that the defined set of parameters is consistent with the existing mobile environment.

The final step consists in the validation that the correct decision has been taken. It should be noted that, except in [52] and [50], this step is never considered in the literature. The main criterion to evaluate the result is restricted to the performance of the decision delay and the computed efficiency being. In real life, the decision can be made autonomously by the selection entity if it is straightforward or corresponds to the cognitive domain of the algorithm. However, it can also involve the human end user and check that he agrees with the decision while giving him a last opportunity to modify it. This step should not be made mandatory, but applied only when the decision triggers a change to an unknown network operator for example.

In summary, the access network selection plays a major part in the control of its connectivity by the terminal. The trend in the research literature is to make the decision at the terminal, based on its own context (applications requirements, subscription rates, security credentials, signal quality...) and sometimes assisted by parameters or policies retrieved from the network. However, most of the algorithms proposed require a continuous execution and thus consume a lot of processing power, which is not convenient in a mobile device with scarce battery resources. Moreover, the decision is never checked against the user final opinion, especially when missing information may have led to the wrong choice. When performed with the network perspective, the objective of choosing the terminal access network is to provide an intelligent access to moving multimode terminals in a wide and heterogeneous network, while saving on the core network load. In that case, neither the user's preferences are taken into consideration, nor the networks which do not pertain to the network operator.

2.5 Self-Organizing and Autonomous Systems

As stated before, the conception of future architectures introduces a totally new cognitive plane, where the environment is sensed and observed, leading to the acquisition of knowledge which is exploited in a novel capability of self-management. Autonomous systems are the natural evolution of expert and knowledge-based systems. They are at the convergence of cognition, information engineering and natural sciences. Cognition implies acquisition of knowledge, through awareness, perception or reasoning. These systems are able to go beyond human and existing machine capabilities because they interact and combine the mechanisms of engineering and natural sciences, such as social behaviour analysis or neurosciences. They provide a high degree of robustness because they are tailored to react to changes in their environment and able to apply human common sense when handling with these situations. These systems are adaptable to cope with unexpected situations and are in continuous evolution at all levels, whether it be data, environment or goals. After a preliminary self-configuration according to initial policies, they enter the loop of a decision-making logical sequence with the following phases: service delivery, sensing of the environment, learning, decision and reconfiguration phase, and then back to service delivery.

In this area, IBM has provided a blueprint [62] including "*fundamental concepts, constructs and behaviours for building self-managing autonomic capability into an on-demand computing environment*". The concepts and architecture proposed in this white paper constitute the leading reference in the domain of autonomous systems [63]. The next section provides an overview of its main concepts and architecture. It is followed by an analysis of how these notions are used in the networking area.

An autonomic system is characterized by an autonomic behaviour, e.g., it has the ability to manage its own operation following some dynamic changes occurring in its environment. This management is performed primarily according to some internal policies and without requiring actions from a human user. The system operates by undertaking control loops. It senses its operating environment, works with models that analyse its own behaviour in that environment, and, based on existing policies, derives the appropriate actions to adapt and change the environment or its behaviour. It provides capabilities (called properties) for self-configuration, self-healing or self-optimization. With the self-configuration, the system adapts dynamically to the changes, using internal policies. The self-healing property is applied to cases where the system reacts to disruptions or to initiate a corrective action by altering its own state. This improves its resilience. The self-optimization feature monitors and tunes the system automatically to optimally meet the end user needs. The main benefits of autonomic behaviour is that it relies on software algorithms which analyse the changes more accurately and cope with the required actions in a faster and more transparent manner, offloading an intervention from the human user. It allows monitoring more easily a larger amount of data.

The basic entity in this architecture is the Autonomic Manager. It automates some management function and delegates this function according to some behaviour defined by its available interfaces. It executes an intelligent control loop, illustrated in Figure 14, which can be split into four parts:

1 - Monitor function: collect detailed information on the environment changes based on monitoring and sensors;
2 - Analyse function: apply control policies to determine and evaluate potential actions;
3 - Plan function: decide on the best suited action,
4 - Execute function: trigger actions according to the decision.

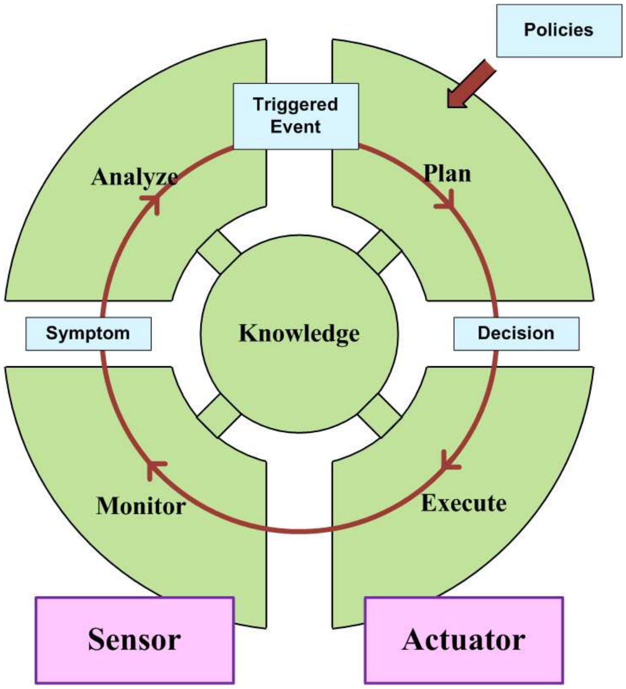


Figure 14: Internal functional view of the control loop

Complemented with the Knowledge function, this architecture is called the “MAPEK” architecture. In fact, it is more a concept than a concrete architecture. However, it makes the whole architecture portable and platform agnostic.

The autonomic system architecture is structured according to the hierarchy and building blocks presented in Figure 15. The next paragraph describes their main roles and how they are combined.

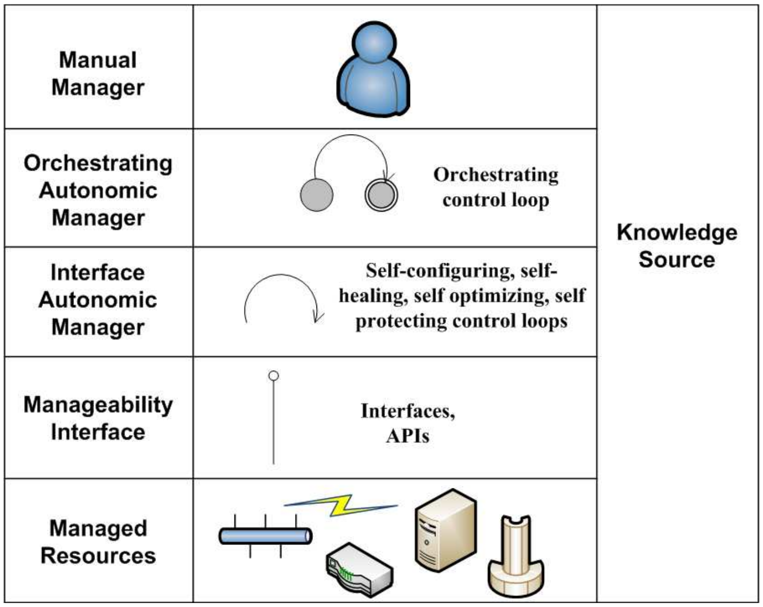

Manual Manager
Orchestrating Autonomic Manager
Orchestrating control loop
Interface Autonomic Manager
Self-configuring, self-healing, self optimizing, self protecting control loops
Manageability Interface
Interfaces, APIs
Managed Resources
Knowledge Source

Figure 15: Autonomic Control Hierarchy

The core component is the Orchestrating Autonomic Manager (OAM). As an autonomic manager, it manages other software or hardware components using a control loop. Its control loop is made around functions for monitoring the system environment, analysing the events, planning corrective actions and executing them. The OAM's role is to work with the other autonomic managers while providing coordination functions. Autonomic managers use policies to govern the behaviour of intelligent control loops. These policies are made of a number of directives which can be derived to specific actions to achieve the system goals or objectives. They are defined by the system administrators. The OAM is assisted by a Manual Manager which provides a common and consistent user interface based on an integrated console. Indeed, in some cases, the OAM cannot reach a sufficient confidence level based on the internal policies at hand. Some tasks and decisions need to involve human intervention, the role of the integrated system being to facilitate the task of the human professionals. The OAM functions using by Interface Autonomic Managers (IAMs) which access Managed Resources through the Manageability Interface, i.e., a set of available interfaces to the instances of resources. Both entities rely on mechanisms such as Log files, events, commands, APIs and configuration files to assess whether the environment requires a corrective action or not. Tuning each service individually cannot provide the same efficiency due to the complexity of the choices to be made. An optimal performance is obtained when some coordination of the policies and actions of all the IAMs in order to provide an ***integrated autonomic behaviour*** exists. The Manageability Interface programs the sensor (detection) and actuator (execution) behaviour for the managed resource, and maps the sensor and actuator interfaces to existing network interfaces. Each autonomic manager uses a manageability interface to monitor and control the resource it manages. Most of these functions are automated to fulfil the user requirements.

For a better tuning of the system and adaptation to the user, a basic knowledge source containing registries, dictionaries, databases, or other repositories according to the system monitored, is installed by the system administrators so that it can be shared by all the autonomic managers. It is further enhanced by self-learning in an evolutionary process through progressive steps, from the human machine interaction, using the Manual Manager.

During the first operations, or in case of an unrecognized event, the Manual Manager is called to provide the solution, which is then stored in the knowledge base. The next time the same event is triggered, a decision is taken which may be confirmed or not by the Manual Manager. After a few similar executions, the level of confidence for that (event, solution) couple is estimated high enough to leave the autonomic system make the decision by itself.

Autonomous systems are still at the embryonic stage, but they recently sparked off a lot of interest and are currently being applied to very different domains such as network architecture, radio networks, sensor networks, knowledge discovery engines or even cognitive modelling [64]. Practical application fields cover Cognitive Radio (CR), Self-Organizing Networks (SON) or autonomic computing [62].

One of the application fields in wireless communications is the use of cognitive systems for optimizing the radio access [65]. Cognitive Radios benefit from the evolution of Software Defined Radios to improve the flexibility of the wireless communications. A CR system senses, observes and analyses the radio environment to detect spectrum holes in terms of time and location, estimates the channel state and predicts its future usage [66], as shown in Figure 16 (copied from [65]). When possible, it triggers the transmission on the available part of the spectrum by secondary users, i.e. users who may not have had access to this spectrum otherwise, adapting their radio parameters such as transmission power or bandwidth occupation to fulfil one major constraint: not generating interference to the primary users. This monitoring of the network is a continuous operation which allows an efficient allocation of available resources and reliable wireless communications between end-users.

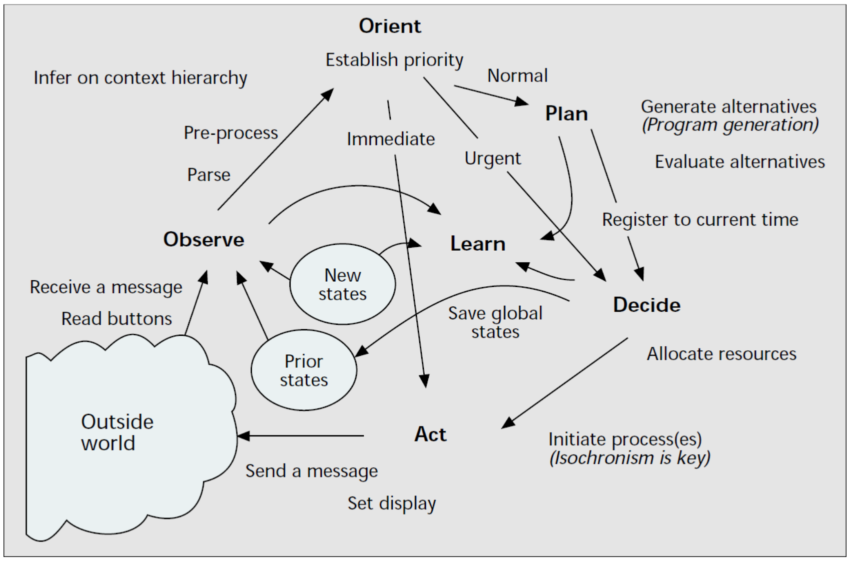


Figure 16: The Cognition Cycle

Self-Organizing Networks are networks which are capable to reconfigure themselves to respond to external context changes. Their main features are self-awareness and auto-learning functionalities, which can be located in entities distributed across the various network components. They are able to dynamically modify their topology and adapt their operational parameters to respond to the needs of the users while enforcing the network policies and improving the performance of the network. The UniverSelf project [67] is defining a Unified Management Framework (UMF) using the autonomic system concepts and capable of managing any type of network. A SON is usually reconfigurable and managed by smart entities, able to update the policies according to the situations encountered. For instance, autonomous resource management [68] improves the resource usage and the user satisfaction when communicating through wireless devices and networks. Applications and devices are then able to exploit the increased network performance while being unaware of the reconfiguration and changes executed in the network. Autonomic computing facilitates the daily management of large and complex networks.

These concepts are introduced in a very basic and semi-empirical way in the existing mobile terminals to decide on which access network the mobile should connect, as described in Section 2.4. By mirroring the self-management architectures currently defined for large networks, it sounds interesting to make an analogy and apply it to the self-configuration of the MT, more particularly to the coordination of the different techniques involved in the solution to our problem.

2.6 Intelligent Transport Systems / ITS Model

The miniaturization of electronic components and the rapid expansion of mobile communications have brought Internet in most of the handheld devices. New vertical applications are being designed to improve our daily lives with added security, flexibility and respect of the eco-system. The future devices will bring to people improved healthcare, living, transportation and energy. Following the same progress, the mobile terminals have become multimode and contain more than one radio interface. They are able to access different types of networks according to their environment.

A typical example of this evolution is the transportation domain. A whole set of technologies and applications is being designed for Intelligent Transport Systems or ITS to enhance the quality of our travelling experience and provide the drivers and traffic authorities with new smart capabilities for road safety, traffic efficiency, local services or internet access. Innovation in this domain has been running for a few years with several research projects such as Sevecom [69] or COMeSafety [70]. Concurrently, standardization ([71], [72]) is setting the framework and rules to enable the compulsory interoperability of the future devices. This new field brings into the communication sub-system a new access technology, the ITS G5, based on the IEEE 802.11p amendment of the IEEE 802.11 standard; but other technologies, such as the Wi-Fi (IEEE 802.11a/b/g/n) or cellular networks (GSM, GPRS, UMTS, LTE and beyond), can be used as well because they are already deployed, they have a larger coverage area in the case of cellular, and they enable the usage of existing personal handheld terminals. More technologies such as digital broadcast (for example Digital Video Broadcast, DVB or Digital Audio Broadcasting, DAB), infra-red, and satellite systems could also be envisioned.

Even though different standardization bodies in Europe and around the world are considering this new domain, a global agreement has been reached to work with a common framework architecture derived from the OSI (Open Systems Interconnection) model. An outline of this reference architecture [73], called the ITS Station (ITSS) model, is pictured in Figure 17.

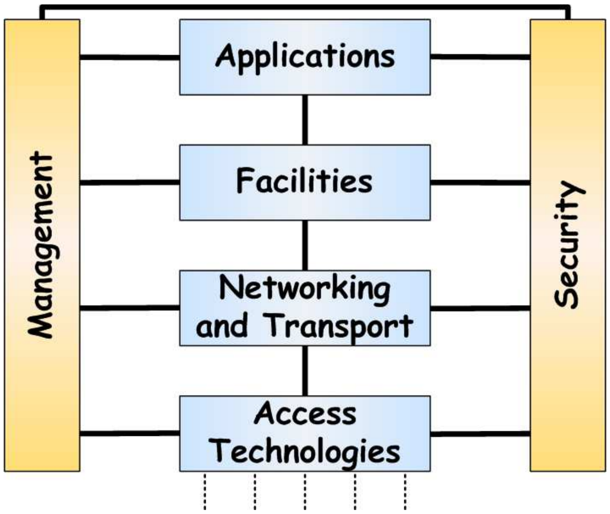


Figure 17: ITS Station Model

The centre part of the ITSS model includes the various layers for the data plane and information transfer. At the top can be found the ITS Applications, supporting the vehicles and traffic operations. Below, the Facilities layer provides the applications with common tools and generic services such as the management of the different messages, cooperative awareness or a Local Dynamic Map (LDM) which maintains a dynamic network topology of the area around the ITSS. Communications are handled by the Networking and Transport (N&T) layer, with specific protocols such as the GeoNetworking (GN), or more usual ones such as TCP or UDP associated to IPv6. The GN is an ITS-specific network sub-layer that uses geographical information to disseminate information and transport data packets. Finally, packets are forwarded to the physical network by the Access Technologies layer, which merges the OSI Data Link and Physical layers. On the sides of the model, the Management and Security layers provide utilities and cross-layer support to the data plane layers for an enhanced operation of the ITSS. Moreover, the architecture considers a varied set of ITSS. They can be handheld terminals, personal devices, cars, trucks, public vehicles such as buses or trams, but also traffic lights, variable message signs, traffic monitoring centres, etc. Some of these ITSS are moving very fast, others are stationary. Different scenarios are defined, according to the communication endpoints (Vehicle -V- or Infrastructure -I-). V2V communications are performed in ad hoc mode between vehicles while V2I and I2V are usually linked with some central servers. In the ITS environment, when operated, the access technologies often involve independent network operators. Highway operators or local authorities will provide the G5 infrastructure along the roads, while mobile telecommunications operators own the cellular access, and gas or electricity companies will offer Wi-Fi hotspots at the gas or charging stations.

The new applications introduced in the ITS environment imply new constraints on the communications sub-system. For example, road safety applications developed to prevent car crashes require very low latency communications between the vehicles, which is achievable only with the ITS G5 access in V2V mode. On the other hand, entertainment applications may require large bandwidths which can be obtained only with Wi-Fi or LTE cellular networks. As a consequence, the selection of the access network to be used depends not only on the radio signal level, but also on the application requirements and on other system parameters such as the source of energy (e.g., engine vs. battery).

Most of this thesis has been developed in the context of a pre-deployment Field Operational Trial (FOT) [74]. The equipment currently implemented to execute this test is configured statically. The same application always uses the same type of access technology, whatever the context of the ITS Station. The control of each network device is performed with specific pieces of software in the Facilities and N&T Layers. The terminal being multimode, each technology or modem brand supported requires the development of extensions to the control software in addition to the specific device drivers, which allows very little flexibility when porting from one environment to the other. The ITS world is thus a typical notebook case where a smart access technology selection algorithm, coupled with a strong level of abstraction for the control of the different access technologies and the support of mobility between independent operators is required.

2.7 Summary

This chapter started with the observation that applications may freeze or stop when changing their access interface. Next, we have presented three technological fields that can contribute towards our target architecture and will help achieve our final objective. Nowadays, there are techniques to handle multimode terminals in an abstracted way, to manage mobility, roaming and network discontinuities and to optimize a smart network access selection. Moreover, techniques to manage networks through an autonomous behaviour have been described, even though they are not yet applied in the field of mobile terminals operation. As outlined in each section of this chapter, the current Internet and the MT architecture limit the improvement that can be expected from each of these techniques. These limitations are summarized in Table 3 below.

Existing behaviours and techniques	**Limitation**
Application session continuity after change of network	A large majority of existing Internet applications are based on TCP connections. When the TCP connection is broken, the application is responsible to handle its re-connection.
Media Independent Handover for heterogeneous networks	Limited to handover functions, network interfaces and control plane. Does not overtake other services: QoS, AAA (Authentication, Authorization and Accounting), battery consumption, positioning…
Mobility Management	The mechanisms proposed require heavy changes in the network. They depend on control entities located in the network that must be owned and maintained by specific organizations. They are not able to support efficiently seamless handover between independent networks.
TCP session is broken when IP address changes.	Packets are lost. Heavy DTN techniques can be used to buffer packets during the connection disruptions,

	but are over-sized (hours/minutes) compared to the requirements of a short handover (in sec.).
Network Access Selection	Available mechanisms are too simple or imply a strong impact on the terminal processing power .
Autonomous systems	They are used for cognitive radio or network self-configuration. They could be applied as well to MT self-configuration to optimize its operation.

Table 3: Current architecture limitations

A vertical application field, the ITS domain, has been presented. It constitutes a typical application case with three access technologies. We argue that the four techniques selected and analysed can be adapted and combined efficiently to achieve our objective, the target scheme implying also a strong level of cross layer design together with enhanced functionalities. We consider here that, for each of the potential access networks taken individually, the terminal is able to monitor the network availability from each of its active interfaces, to configure a network interface while performing the required security authentication and to help the application recover from a TCP connection failure. In summary, techniques and technologies are available, but they are often applied individually. Their efficiency could be improved by combining them in a single framework. All of these techniques affect the mobile terminal which is the only node that the end user can control. 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.

In the next chapter, some typical scenarios will be devised, allowing the identification of the main requirements to be applied to the target architecture.

CHAPTER 3 - SERVICES AND REQUIREMENTS

The preliminary work performed during the first part of this thesis has shown various situations where the optimized control of the connectivity of a multimode terminal would be beneficial if not mandatory. A set of technologies has been selected to provide a solution to these issues. It targets a cross layer approach of the system architecture, an abstraction framework to hide the technology specificities and an autonomous behaviour in order to achieve the optimization of the mobile network access. In this chapter, three typical scenarios based on this work are devised. Their objective is to identify use cases where the MT will not behave according to its user's needs. These situations may imply point-to-point (unicast), but also point-to-multipoint (multicast or broadcast) communications. Next, some requirements for the target architecture are derived from the use cases. Finally, some selected use cases are reproduced by simulation, in order to validate the problem and constitute a reference for the following evaluation of the target framework.

3.1 Determination of Typical Scenarios

In this section, some typical scenarios are proposed, with the objective to derive the requirements, technology blocking points and challenges to solve in the remaining part of this work. This study targets multimode mobile terminals which offer the capability to simultaneously or sequentially attach to various and heterogeneous access networks. The first scenario showcases a mobile terminal moving between independent networks and addresses the problem of the continuity of the application sessions, especially when the application is based on TCP connections. The second scenario features a car device which needs to access simultaneously the operated cellular network and the non-operated ITS environment. The third scenario shows a handheld device in the case of a public emergency situation such as an earthquake or a tsunami. The primary notification is received on a network still available, providing the data necessary to start listening to the follow-up information using a dedicated application and a more suitable access.

<u>Mobile session continuity</u>

The first scenario considered is the case of a smartphone user that moves out of the coverage of a private network, either home or office network, and needs to execute a smooth handover to the cell of his mobile operator network, even when running an application based on the TCP protocol. This is illustrated in Figure 18 with a picture of the scenario setting. Pierre is leaving the university with his colleague Paul. Both are reporters for the local newspaper. Pierre is watching an important keynote on the web with his smartphone. This application is built on top of the TCP protocol. However, they cannot delay their departure because their chief editor asked them to attend the daily red carpet showcase at the Cannes Film Festival. While Paul drives out of the parking lot, Pierre's smartphone loses the university Wi-Fi network and smoothly switches to the 3G network. This is transparent for Pierre, but his device has been able to handle all the required parameters to maintain the application sessions open: network address allocation, security authentication, network availability and monitoring.

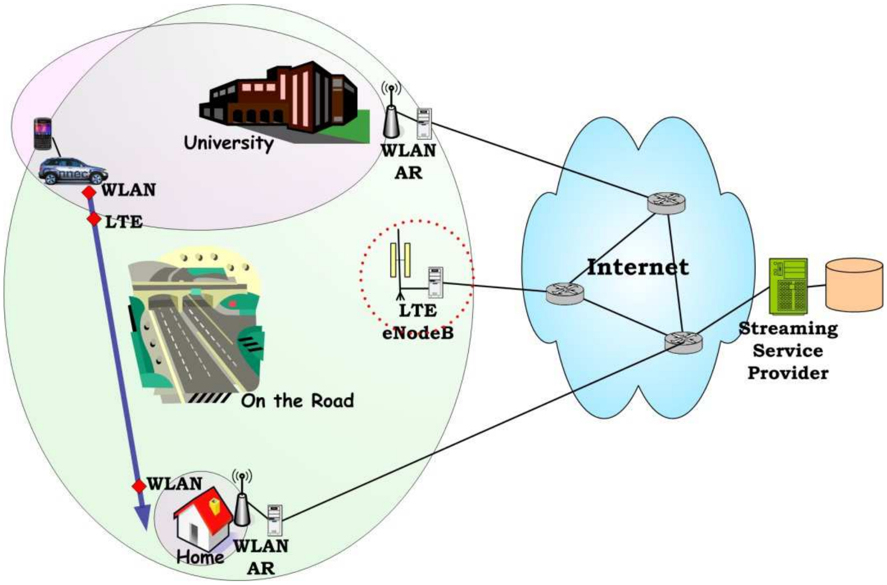


Figure 18: Mobility between independent operators

Indeed, the same smooth transfer is performed by his terminal when they briefly stop at his home to fetch his press accreditation card. His home network is recognized, and as a preferred network, all his data connections are moved to the new network.

A similar use case would occur when the application or the user wants to control the attachment of the device at the access point level, based on retrieved network information. Motivation for such a decision could be due to the prevention of battery consumption, network congestion avoidance, application data rate balancing, subscription charging or network traffic efficiency.

Vehicular communications

The second scenario takes place in the vehicular vertical application domain. The car application in an ITS Station, or ITSS, may decide at some point to handover from the road operator network to a mobile operator network with higher available bandwidth to receive traffic information geo-broadcasted or resume downloading some media faster. In Figure 19, the ITSS located in Paul's electric car has activated its ITS G5 access to be tuned to safety messages notifying road hazards. As he enters the city suburbs, the car monitoring system detects that the batteries are getting below a pre-defined threshold and need re-charging. The POI (Point of Interest) notification message advertising charging spots is disseminated both by a nearby roadside station on a G5 channel and on the cellular network. Based on its configured preferences, the monitoring application decides to switch to the cellular because the message content covers a larger geographical area and contains a wider range of charging stations, thanks to the larger bandwidth. The system is thus configured to start listening automatically to the cellular access network when instructed to do so by the application, without disabling the road hazard monitoring executed by a collocated application.

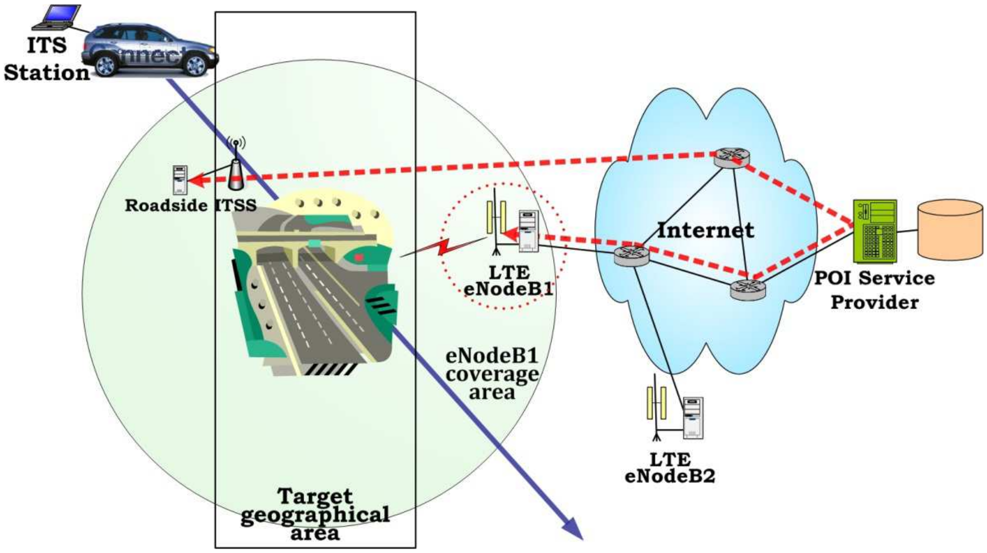


Figure 19: Geographical message dissemination

Another usage in the vehicular domain would be when Paul arrives at his contracted charging station where Wi-Fi is available for free as part of his subscription. His system switches his entertainment data traffic to that technology as soon as it is detected and enables him to start downloading a short movie while the battery is charging.

Emergency notifications

A third use case that may be considered is the reception of an emergency public warning in case of a major disaster [1]. Public Authorities are working jointly with operators to benefit from the upcoming broadcasting technologies and progress beyond the capabilities of the former audio sirens. Broadcast-capable networks based on satellites, mobile wireless cells (3GPP MBMS, Multimedia Broadcast/Multicast Service), video or audio broadcasting, amateur radios or the Internet, are currently being adapted for this usage. For instance, in the cellular systems, the MBMS is an enhancement which provides a point-to-multipoint capability for Broadcast and Multicast Services, allowing resources to be shared in the network [75]. It supports two modes of operation: the Broadcast mode and the Multicast mode. The objective of the MBMS is to transmit the data from a single source entity to multiple recipients at once, thus saving wireless resources. The warning system [76] is planned so that the user receives a primary notification through one of his available connections. He can then switch to a secondary notification system when starting a specific application able to receive follow-up information such as text, audio or graphical data and be instructed what to do and where to get help.

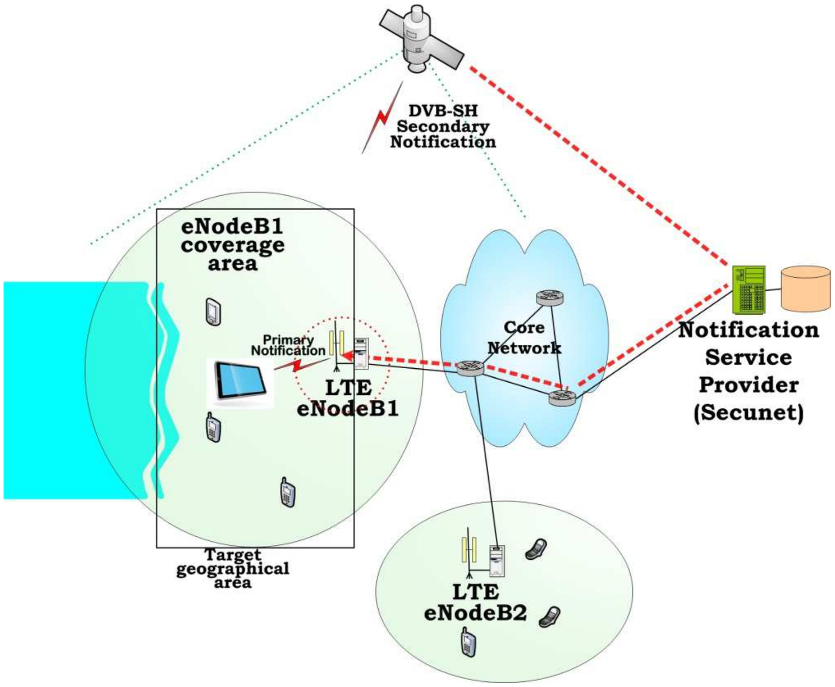


Figure 20: Emergency Public Warning

Figure 20 illustrates a tsunami primary notification delivered through the cellular network, while a secondary notification including maps and video recommendations is provided through a DVB-SH (DVB for Satellite Handheld) satellite system in that area. Justin, the mobile user, is connected with his tablet when he receives the primary notification of an upcoming tsunami. He is able to select the best available network to receive the secondary notification and get more information, not only in terms of signal, but also considering the availability of the information or the language used. The requirements leading to this network selection have been fine-tuned when he installed his application or dynamically inserted in the terminal after the alert notification. He runs out of his office, while his applications are still running to receive the safety recommendations. His system is able to transfer them automatically to the networks which are still operational.

3.2 Specific System Requirements

From the analysis of the previous scenarios, a first set of requirements has been obtained. In all the scenarios, the MT is the main actor. Accordingly, the main constraint of our scheme is that only the MT should be affected, either by the modification of its internal entities or by the introduction of new entities. The mechanisms defined must remain independent of the wireless access technologies and network providers, while still complying with the constraints and existing policies that a specific network may impose on the device. The “mobile-only” requirement can be also justified from a market point of view, considering that the average life cycle of a mobile device is around two to three years, with a development cost shared by millions of users, while the cost of upgrading the network infrastructure is very high and mostly supported by the operators. This may run counter to the legacy technological trends which were driving mobile networks until now. As shown in Chapter 2, they avoid the usage of abstraction layers and put the control at the network side. However, it goes along the emerging user-centric tendency.

The proposed concept must enable an always-on connectivity and a smooth roaming from one network or domain to another. It must guarantee a simple usage of the device, hiding the complexity of the connectivity issue to the human user. The abstraction concept of the MIH Services, as proposed by the IEEE 802.21, together with their associated mechanisms, which aim at optimizing the operation of network services across heterogeneous access technologies and systems, must be extended to cover any type of connectivity-related service involving a network interface or any other device present in the terminal. An access network selection algorithm must be implemented to optimize the performance of the system. It is based on static and dynamic information about the underlying network capabilities, status and performance and on a wider range of criteria involving the whole device and its context. Since no change is expected in the network, and to ensure that the mobile can attach seamlessly to any type of network, existing mobility schemes must be enabled and used transparently inside each operator domain. Moreover, other mechanisms must be designed to enable a fluent mobility and session adaptation between operator-independent environments, nevertheless complying with the same requirement of surviving the change of the IP address without interruption of the user application. Generic service enablers should be developed which, for each key service including access network selection, connectivity and session management, provide solution-independent facilities to optimize the network operations. In order to provide a wider flexibility of the system, it must be possible to introduce new technologies and service enablers without breaking the existing architecture.

To maintain a smooth operation, self-decision and reconfiguration should be executed whenever possible, while the user can still request status information and change the settings of his terminal at any time, using an interface as simple and as intuitive as possible, increasing his QoE. This could be achieved by mirroring the current developments in autonomous systems in the control of the terminal. These features should be completed by a shared local knowledge source, storing and providing access to standardized policies, cross-layer metrics and system parameters. To ensure backward compatibility and large adoption, this scheme must remain compatible with existing applications and network protocols. Even though networking security issues are not at the core of this thesis, the system should take them into account and not break existing protective mechanisms. A final constraint is to minimize the amount of processing power and energy consumption induced by the designed framework. It must remain as light as possible.

As in any mobile-related mechanism, the main success criterion is the minimization of the number of bytes and packets lost. Here, it signals more the efficiency of the MT operation than the performance of the system in terms of throughput. Additional metrics can be defined to fine tune the system and validate a sound behaviour: switching time between two technologies, total cost of the communication with the objective to use as much as possible the cheapest technology, suppression of application failures when the TCP session is broken, number of handovers and duration of a connection between two handovers to avoid a Ping-Pong effect.

3.3 Reference Simulation Model

In order to provide a reference model for the evaluation of the proposed solution, a simulation setup has been created. It reproduces the behaviour observed during the applications experimentation of Section 2.1 and runs some mobile terminals in a network set-up similar to the scenarios defined in Section 3.1. The simulation scenario and system implemented for this study are described with more details in Section 5.1.3.

Figure 21 shows the network layout used for the simulation. The core of the network is represented by a router, the *net*, which simulates the Internet. It connects an application server, the *srv*, and several access routers, either for a simulated LTE access (see Section 5.1.2), *arLTE*, or for a WLAN access, *arWlan*. The LTE cell has a global coverage and provides an always-on access, while the WLAN availability is restricted to the inside of the circle shown in the picture around each access point (*ap*). Two mobile terminals, *host0* and *host1*, move randomly across the simulation playground. They run one out of three data transfer cases. A Ping test from the mobile to the server evaluates its connectivity. A file transfer application based on TCP simulates periodic requests from the terminal answered by the server. A web browsing application also based on TCP reproduces the behaviour of a user sending requests to the server and waiting for its answer to send the next one. More details on the manner these applications is provided in The application data rate and the simulation duration are set to a fixed value in order to obtain comparable results, as explained in Section 5.2.1. Table 4 shows some statistics on the number of handovers and the average time spent on each technology. Due to the random mobility model adopted and the size of the playground, the average time spent on the WLAN cell is about a third of the total connection time (equal to the simulation time) and the average frequency of handovers is high to demonstrate later the efficiency of the model.

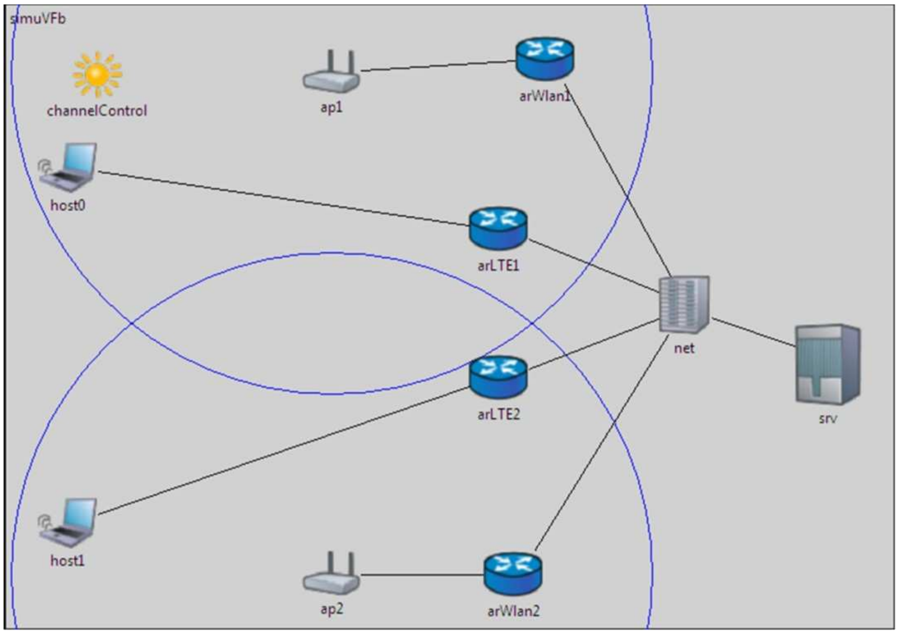


Figure 21: Simulated network scenario

	Min	Mean	Max
Number HO	0.00	17.54	34.00
Time WLAN	24.67	699.23	1999.61
Time LTE	0.00	1300.38	1974.95

Table 4: Statistics on MT mobility

For this evaluation, the standard wireless host model is used and can connect through one of the available access networks: Wi-Fi, and the simulated LTE cellular access. Measurements and statistics have been collected in four different use cases, summarised in Table 5.

Use case Name	Scenario
WLAN 1	Stationary MT with WLAN
WLAN2	Moving MT with WLAN only
LTE	Moving MT with LTE only
CMGR	Moving MT including a standard CMgr

Table 5: Evaluated use cases

In the first use case, the terminal is stationary and uses the WLAN interface. In the second, the mobile moves across the simulation playground using only its WLAN access. It thus loses its connectivity when it gets out of reach of its access point. In the third, the mobile uses the LTE access only; the access is never lost but introduces strong constraints on the data traffic. In the fourth case, the terminal contains a standard Connection Manager and behaves as a smartphone, connecting to the WLAN whenever possible and to the LTE otherwise.

For each simulation run, the following statistics have been collected:

- number of Ping Echo requests transmitted and Echo replies received back by the mobile,
- number of bytes transmitted and received by the applications,
- time when the last packet was sent / received at the MT and in the server with the FTP application,
- number of session establishments and disruptions with the web browsing application,
- connection time to each technology and in total in the last use case with the Connection Manager.

Figure 22 shows the result of the Ping session (lost Echo replies) in each use case for the host. In the WLAN1 and LTE use cases, the terminal is under coverage of the wireless cell for the whole simulation, no packet is lost. When the terminal moves and is restricted to WLAN, its working area is partial and a large number of messages are lost (use case WLAN2). This is improved when the Connection Manager switches the connectivity between the two links, even though some losses can still be observed. The measurements show a loss comprised between 3 and 7 packets for the time of the simulation, with an average of 3 packets

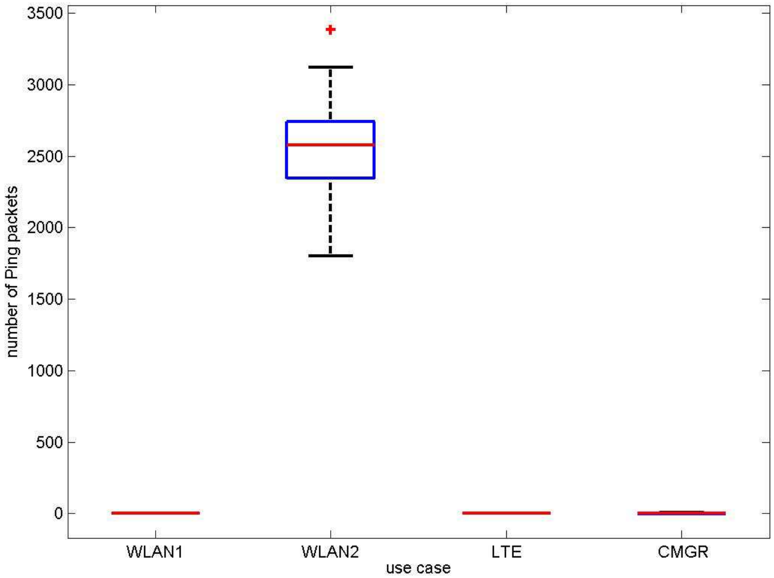


Figure 22: ECHO REPLY loss during the Ping Test

Figure 23 compares the number of bytes received at the destination for the uplink and downlink traffic respectively when using the FTP application. Figure 24 performs the same comparison with the web browsing application. In use cases WLAN1 and LTE, all the traffic is received. In use cases WLAN2 and CMGR, the FTP application is able to transfer data only until the first handover. Afterwards, the TCP retransmission timer expires its number of retries; the session is broken and cannot be recovered. The web browsing application, on its side, is less impacted because it keeps starting new sessions. When a session has failed, it tries to open a new TCP connection. Its success rate is higher in use case with CMGR than in use case WLAN2 because the LTE network can be used as well, so the number of bytes transmitted is higher.

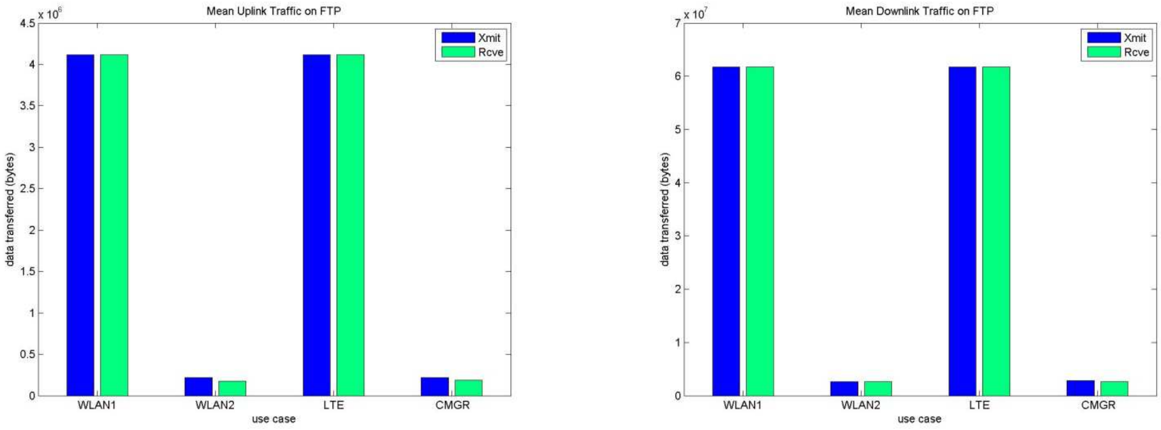


Figure 23: Comparison of traffic during a FTP session for uplink (left) and downlink (right)

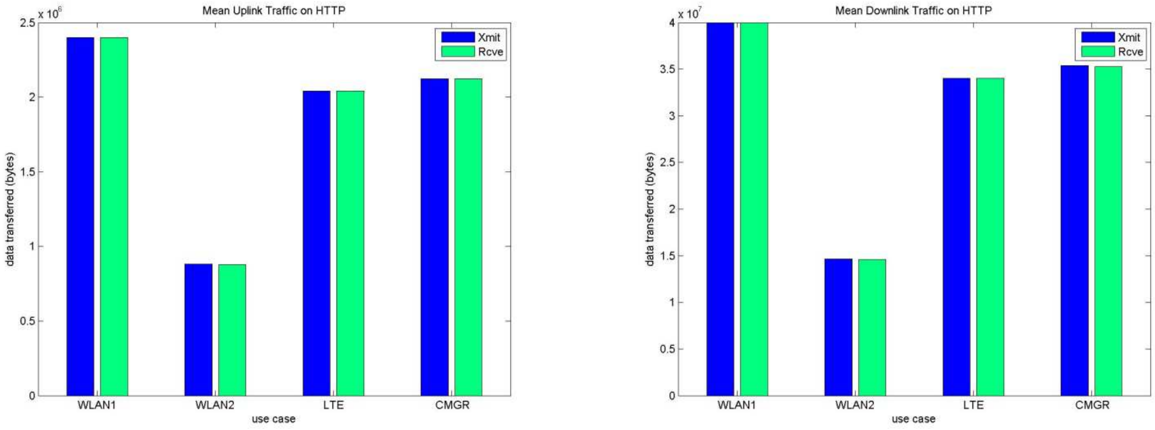


Figure 24: Comparison of traffic during a HTTP session for uplink (left) and downlink (right)

Figure 25 shows the number of broken sessions according to the total number of opened sessions for the web browsing application. In use cases WLAN1 and LTE, all the sessions are successful, the number of sessions broken is equal to zero. Because of the lower bandwidth and higher delay in the LTE network, a smaller number of sessions can be established in the same allocated testing time. In use case WLAN2, the total number of sessions is higher than in use case WLAN1 because the MT retries many times to establish its sessions when it is not under coverage of the WLAN. Overall, this is the case when the data traffic is the lowest. Several sessions are broken while they are executing. Again, in use case CMGR, this situation is improved, more sessions can be established. However, some of them still cannot end successfully.

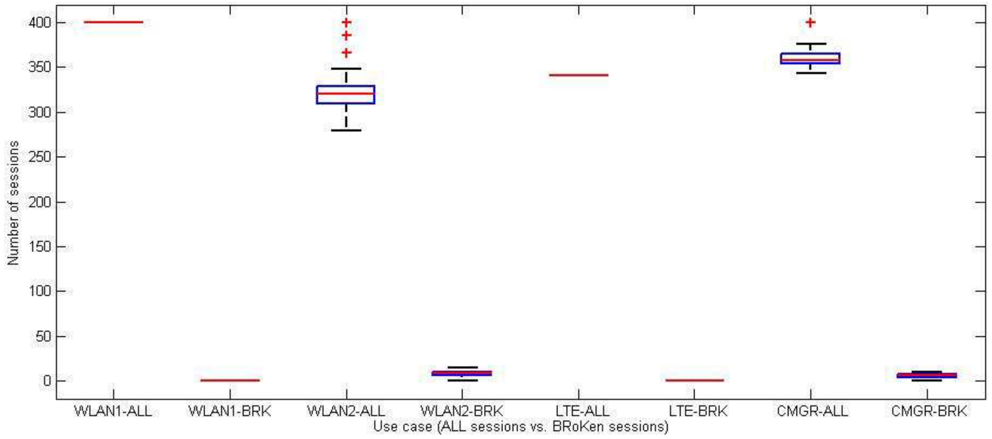


Figure 25: number of HTTP sessions broken vs. total number of sessions

Figure 26 and Figure 27 show for the different nodes (terminal and application server) when the last FTP packet could be sent and was received. It is clear that in use cases WLAN2 and CMGR, the traffic stops right after the first handover and is never recovered.

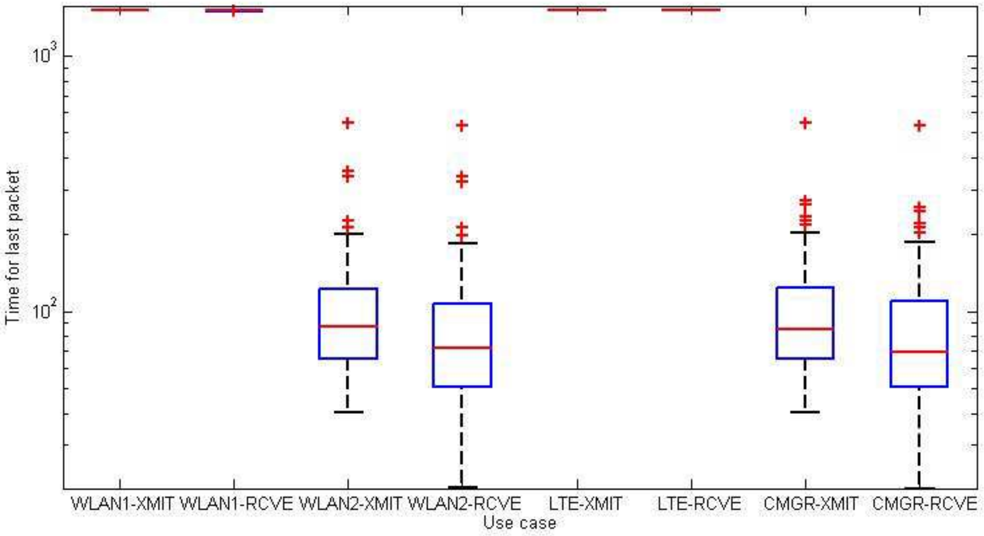


Figure 26: Last packet transmission and reception time at MT

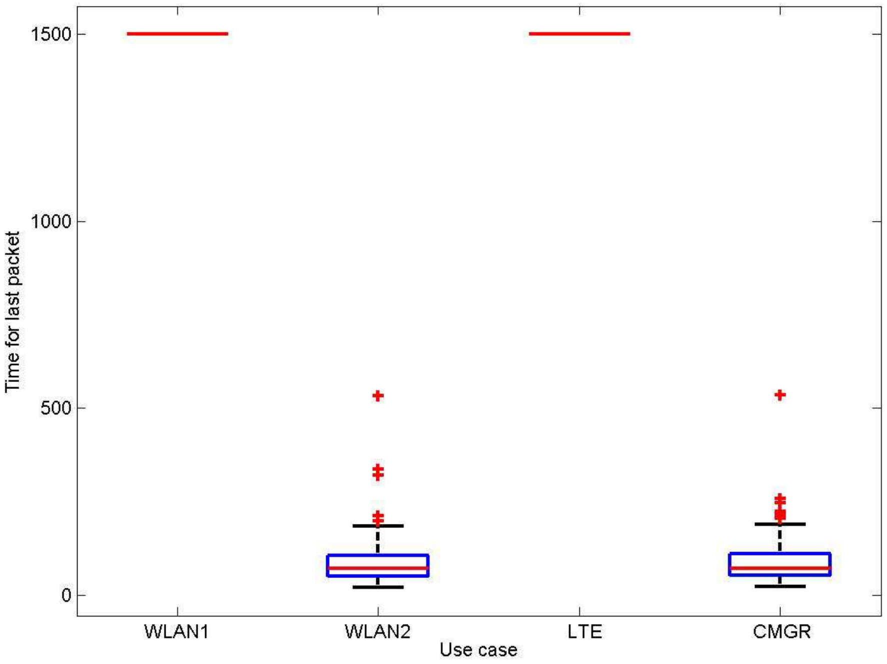


Figure 27: Last packet transmission and reception time at server

In summary, the simulation results in this section reproduce the mobile and application behaviour observed in real life. A synthesis of the observations is provided in Table 6. In two use cases (WLAN1 and LTE), the mobile remains under the coverage of the same access technology and thus experiences no transmission problem. In use case WLAN2, the mobile can communicate only when in the WLAN cell. The Ping test shows an important loss rate. After the first handover, the FTP session is broken and cannot recover, so its traffic is stopped completely. The web browsing application based on HTTP cannot complete as many sessions and several of them are broken midway through their execution. The situation is improved for the Ping test and the web browsing application when a Connection Manager is introduced in the terminal for use case CMGR. However, the result is not yet optimal and the problem encountered with the FTP application is not solved.

Use case Name	Ping test	FTP test	HTTP test
WLAN 1	No loss	No loss	No loss
WLAN2	Many packets lost	Loss without recovery	Loss with recovery
LTE	No loss	No loss	No loss
CMGR	A few packets lost	Loss without recovery	Loss with recovery and higher traffic rate than in WLAN2 and LTE

Table 6: Traffic observations according to the application

Based on these results, the case when using the always-on LTE access looks like a very good solution. However, parameters other than the signal level and the network availability must be taken into account: available bandwidth and data rate, communication cost or battery life among others. For these parameters, the LTE access still imposes stringent constraints. Moreover, the always-on hypothesis adopted here is slightly optimistic compared to real life. Accordingly, it seems necessary to revise the Connection Manager strategy to be able to solve these issues.

3.4 Summary

The scope of the thesis study has been outlined in this chapter. In its first section, three typical scenarios showcase the issues raised in Chapter 2. The first one addresses mobile session continuity. It presents a use case where a mobile uses switches from a campus to a cellular network, finally ending in a well-known private network. The second scenario addresses vehicular communications. It shows the ITS Station in a car which has to enable connectivity to at least three access networks simultaneously and additionally receive information from various devices such as the positioning system or other monitoring sensors. The third scenario addresses emergency notification. It proposes a use case where a mobile user receives a public warning notification on one of the available networks and starts an application that mandates another network to receive follow-up information.

Based on these scenarios and on the analysis performed in Chapter 2, a set of requirements has been determined. The main requirement is to modify only the mobile terminal, leaving the network totally unaffected. Connectivity has to be maintained efficiently while remaining transparent to the applications. The terminal usage must remain simple and the overhead on its processing performance and battery consumption kept at a low level. From an internal perspective, the system should capitalize on the abstraction model introduced in the MIH standard. Services like a smart access network selection or session continuity should be included. The system should also capitalize on the layered architecture introduced in autonomous systems, 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.

In order to assess the issues observed on a real terminal and produce a reference for the evaluation of the final system, a simulation model has been set up that displays mobile nodes communicating in a heterogeneous environment. Several use cases have been tested with a WLAN stationary connection, then mobile WLAN or LTE, and finally introducing a Connection Manager which fosters the WLAN connectivity when available. Since the test performed in Section 2.1 has shown that applications based on TCP connections are the most sensitive, two applications have been used: file transfer and web browsing. They are supplemented by a Ping test which indicates the level of connectivity in a specific test. The simulation results clearly show that the file transfer application is stopped after the first handover, while the web browsing application is able to recover, but encounters several session breaks, thus reducing its overall efficiency. This is confirmed by the Ping test which suffers losses even when a CMgr is introduced in the terminal.

These are the issues that will have to be solved in the next chapter, taking into account the requirements defined in Section 3.2 and using the scenarios defined in Section 3.1.

CHAPTER 4 - THE CONNECTIVITY CONTROL FRAMEWORK

The previous chapter has defined scenarios where multimode devices need to roam seamlessly and independently of the network provider between heterogeneous environments. The requirements that the target architecture should meet to achieve this goal were derived from the technology analysis of Chapter 2 and these scenarios. In particular, the option was chosen to modify the mobile terminal only and keep the network entities untouched. This chapter proposes a solution based on a cross-layer architecture that leverages the optimization of some 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 multi-homing management, and application session management are combined and enhanced to reach this objective. Moreover, the new cognitive capabilities of autonomous systems are involved to bring autonomy to the roaming and support a faster decentralized operation. In this chapter, starting from the timeline of the functions involved in the operation of a mobile terminal, the main building blocks are identified and organised in a global layered architecture, the Connectivity Control Framework (CCF). Its behaviour is analysed from an operational point of view, followed by a highlight of the services it provides to the other entities located in the mobile itself and in the network. Its internal and external interactions are described, which leads to a closer description of the framework internals. Each layer or component is presented from a functional and operational point of view, showing its optimizations, what it brings to the global system and how the various entities cooperate towards the target objective to connect efficiently the mobile terminal independently of its context.

4.1 General description

This section provides a global description of the framework developed in this thesis. This description starts with a short analysis of the networking functions required to successfully operate the connectivity of a multimode terminal. It is followed by a high-level description of the architecture of the CCF, introducing each of its components and their role in the whole system. This presentation is completed by the presentation and analysis of a set of generic scenes, extracted from the scenarios defined in Section 3.1 and which intend to demonstrate the global operation of the system. These scenes address simple operations. Before its initial operation, the mobile terminal is configured. The real connectivity operation begins when it is started and has to attach an access network. Later, an application is launched, which triggers the setup of its session, followed by the transfer of data packets. The application or the connectivity may have to face events which require inter-domain mobility or multi-homing.

4.1.1 Functional View

In order to be able to fulfil properly its objective, a mobile terminal should be able to start, attach to the best available network, exchange data with a Correspondent Node, move between its heterogeneous accesses when required and finally, stop all these operations [34]. The following timeline shows a basic sequence of the various functions involved in the terminal in that aim. These functions are triggered by the occurrence of the events shown in the square arrows: start or stop of the MT by the user, start or stop of an application again by the user, detection of an event from the mobile environment.

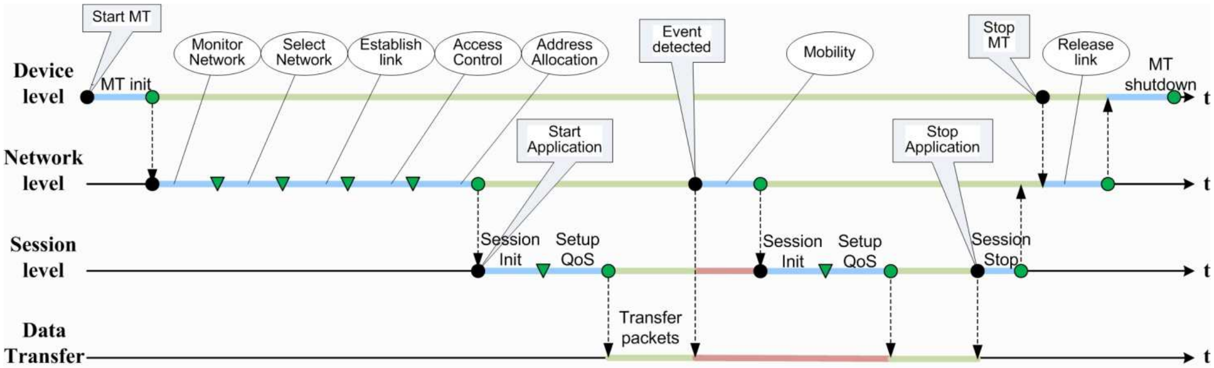


Figure 28: Timeline of mobile terminal operation

These procedures imply that the following set of services is available in the MT:

- Terminal initialization: activate all the software components and bind the various interfaces;
- Network monitoring: monitor dynamically the status of the signal received at the network interfaces;
- Network selection: make the decision for selection of the best available access network;
- Link management: manage network interfaces for link operations;
- Network security: successfully execute registration, access control and authentication;
- Identification management: handle the various addresses and identities of the terminal and of its internal entities;
- Events monitoring: identify events in the connection and in the terminal;
- Session initiation: start and manage each individual session opened by the applications;
- Resource reservation: request the QoS necessary for data exchange;
- Packet transfer: enable packets to flow internally from the application out to the selected

network interface and backwards;

- Mobility management: keep track of current location and handle procedures related to change of location / network attachment, while preserving the opened sessions;
- Terminal termination: unbind the various interfaces and deactivate all the software components.

4.1.2 System View

Figure 29 shows the functional architecture of the CCF. Starting from the existing MT shown in Figure 12, several components have been inserted in the framework to fulfil the requirements listed in Section 3.2, e.g., an abstract interface, generic services, an autonomic manager and a cross-layer database. The framework replaces the Connection Manager. It has been split into several layers, following a pattern similar to the MIH reference model (see Section 2.2). Some of the components shown in the figure are present in existing terminals and remain unchanged. They are pictured as hatched blocks in the figure and include the applications, the Networking Services (NS, e.g., existing handover, security mechanisms, network statistics and parameters management), the TCP/IP protocol stack and the wireless accesses. In this study, 3GPP LTE for the cellular system or WLAN for local access have been selected as wireless technologies, but the system is flexible enough to accommodate any other type of interface.

The lower layer is made of the wireless technologies which offer an access to the mobile networks, whether operated or not. Additionally, it includes other managed resources such as a GNSS device for obtaining the mobile location and time information, the mobile power supply, or eventually one or several specialized sensors (e.g., the gyroscope included in some existing smartphones). None of these components is expected to be modified to satisfy the requirements of the CCF. Each of them is associated to an intermediate component, here named the Link Interface, which provides the managing interface to the CCF internal entities and translates CCF primitives into procedures understandable by the device. This layer offers two interfaces on its upper side: one to the TCP/IP protocol stack on the data plane, unchanged, restricted to the wireless accesses, and a second one to a Media Independent Services Function (MISF) on the control plane. The Media Independent Services (MIS) provided here constitute an extension of the 802.21 MIH Services, in the sense that they cover the operation of both the network interfaces, including additional functions for *resource allocation, multicast selection, load and priority management,* and the other devices. They hide their specificities to the upper layer entities. Another new feature of the MISF is its interaction with the Local Information Base (LIB) to store and retrieve the data related to the lower layer, giving it more intelligence than the simple relay function of the MIH model. As already mentioned in Section 3.2, in this study, the MIS are restricted to the local operations executed in the MT only.

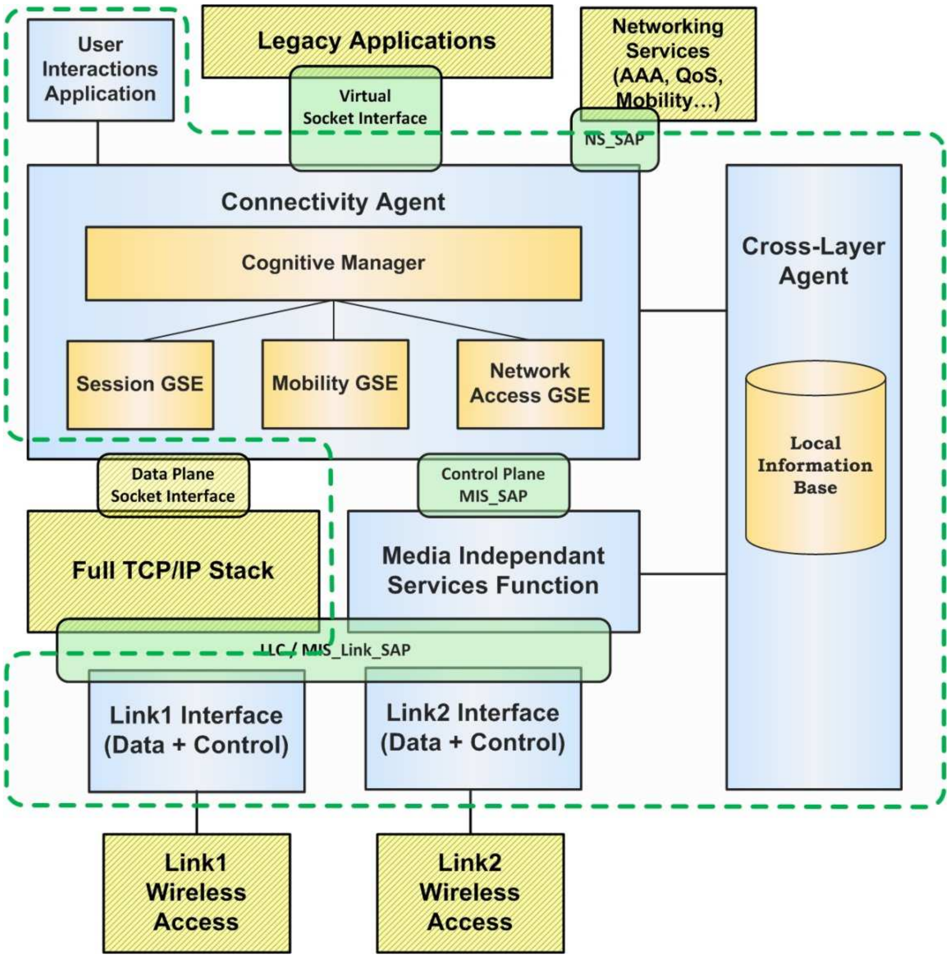


Figure 29: Global architecture of the CCF

Above these building blocks, the picture shows Connectivity Agent (CA), which is the central component of this framework and acts as the system autonomic manager. It is divided into Generic Service Enablers (GSE) and a Cognitive Manager (CM). The meta-functions provided by the GSEs correspond to the main steps of the network connectivity control for a MT. They are directly derived from the technologies presented in Chapter 2 and provide services to the applications and to the NS components which execute the standard control functions (mobility, QoS, AAA, configuration…) mandated by the policies of each connected network. The GSEs cover the selection of the access network, the support of the mobile connectivity and the management of each session, from its setup to the provision of credentials, helping to maintain the roaming transparent to the application. The Network Access GSE (NAGSE) is able to identify the best available network at a given time. The Mobility GSE (MGSE) is able to analyse whether the environment conditions have changed enough to trigger a handover and to connect to the selected network. The Session GSE (SGSE) is capable of dealing with the security session establishment, the address and socket port mapping, the multicast session trigger or the recovery from a TCP connection break, applying light DTN techniques. Should a new service become necessary, the framework is flexible enough to accommodate it with minimum effort, by simply inserting a new GSE in the system, as will be shown in Section 6.2. The control part of the system described so far has been improved in terms of efficiency and operation by inserting at its top an autonomic manager which ensures a smooth functioning of the whole framework. Hence, the GSEs are orchestrated by the CM which acts as a local controller. It acquires knowledge by learning and recording the various steps of an access selection or mobility scenario, and is able to exploit it afterwards to enable its own operation. When facing unknown situations, it has the possibility to request information from the human user through a specific user interface, the User Interactions Application (UIA), shown at the top left of Figure 29.

The CA and the MISF are jointly assisted by the CLA, which supports the cross-layer operations and maintains the LIB. This component is responsible for gathering the data from the wireless access technologies (via the MISF) and from the upper layer entities (via the CM and the GSEs), for aggregating them in the LIB and providing them on request when needed. It contains parameters related to the running applications, the GSEs, the MT internal features and capabilities, the network interfaces and access networks status, the identities of the operating entities and interfaces (e.g., port number). These parameters are completed by a set of policies to govern the behaviour of the system.

In the data plane, techniques like virtual sockets, the use of a personal address or light DTN mechanisms hide the mobility and network changes to the application, which is then able to operate in a manner identical to what it would be in a fixed environment. Above these processes are the user interface and the running applications. The UIA is queried when a human intervention is needed, with a range of functions covering configuration of user preferences to runtime decision making or validation. Finally, the top layer is made of the legacy applications, totally unmodified and kept unaware of the changes of their environment thanks to this generic system.

The proposed framework is thus distributed over several components and greatly improves the performance of connectivity related services by keeping updated information about the events occurring in the managed devices or the available networks and making them available to the upper layers in an abstracted manner. It is embedded in the end-user terminal and introduces several specific new interfaces with different and complementary roles.

- A MIS_LINK_SAP (Service Access Point), to the network access and other device drivers, allows interaction with the Link and Physical layers of wireless interfaces and hardware of other devices.
- An MIS_SAP manages the interactions between the MIS Function and the GSEs.
- In the data plane, a Virtual Socket Interface intercepts the packets from the application and maps their port number and network addresses to those of the real socket, thus hiding the framework operations to the legacy applications.
- The NS_SAP operates in the control plane between the standard Networking Services and the CA.
- Finally, two specific interfaces allow the CCF framework to interact with the mobile end user and the CLA.

An additional interface exists between the Connectivity Agent and the TCP/IP protocol stack. It is realized using the standard Berkeley Socket API.

In the figure, the data path goes from the application to the SGSE, the TCP/IP protocol stack, the MISF with the Link Interface and ends in the link Wireless Access. The other components are part of the control plane, some parts of the CCF operating both in the data and in the control plane. This framework has also been designed trying to mirror the layered model of the autonomous systems architecture described in Section 2.5. This is exposed in details in Section 4.3.4.

4.1.3 System Operational View

This section explains how the framework operates, including the interactions between its components, in a selected set of generic scenes. They show how to compose the various components of the CCF so that they cooperate efficiently towards the goal of the framework. For these scenes, three different types of network domains are considered:

- Domain A is a WLAN private network without any specific mobility support. It may be the user's private home or a coffee shop hotspot. Conceptually, if only IPv6 were used, it would not prevent the use of MIPv6 and the binding of the terminal HoA to the local CoA allocated in this domain. However, this cannot apply to the case of IPv4 (there is no Foreign Agent in this local access network), and would not be compatible with other mechanisms such as PMIP. So we assume here that no mobility mechanism is executed while attached to this domain. In this network, the requirement for authentication is detected when scanning the available networks. Authentication is performed at Layer 2 when the mobile attaches to the Access Point. The IP address is configured afterwards. Here, the mobile owns a global address for each interface, obtained with mechanisms such as DHCP or IPv6 auto-configuration. When IPv6 is used, the auto-configured address is often made of a locator (usually a prefix obtained in the Router Advertisement [77] message) and an interface identifier (usually the IEEE's 64-bit Extended Unique Identifier (EUI-64) address).

- Domain B is a WLAN campus network equipped with some mobility support, such as MIP or PMIP. Authentication to the network is performed at network level after the link attachment and the allocation of the IP address. It is detected via a testing http/IP session and requires credentials such as a user Id and a password. Since mobility is enabled, the terminal handles two addresses: a CoA allocated in the same manner as in domain A and a HoA made of the home prefix and the EUI-64 address, or attributed manually. In the case of PMIP, only the first CoA allocated is used across the various network interfaces as long as the mobile remains in the same domain.

- Domain C is an LTE cellular network and uses 3GPP specific mobility only. Authentication is performed via the SIM (Subscriber Identity Module) card at the beginning of the attachment and before the IP address allocation phase. The network address is allocated when the default bearer is activated. The same address is used for the subsequent dedicated bearers. Here as well, the address allocation mechanism is based on DHCP and IPv6 stateless address auto-configuration, with the difference that it is an entity in the operator network, the PDN-GW, which acts as the DHCP server, allocates a unique IPv6 prefix and interface Id from a pool to each terminal and forwards them through the MME to the mobile, which is then able to construct its IPv6 address.

For each user application, there are two addresses used by the CCF:

- a Personal Address or PA, attributed to the application when it starts. This address is kept identical throughout the whole session;

- an Environment Address or EnvA, which is the address seen by external network nodes and depends on the user network location. It is the global address in the connected network. In case Mobile IP is used, the EnvA is equal to the CoA.

Figure 30 pictures the network layout used for the analysis of the different scenes. Each of them is a small subset of the scenarios presented in Section 3.1. In the figure, the mobile named "*MNxx*" shows the location and movement of the MT in the corresponding scene. Terminal configuration, start-up and the launch of an application occur in all three main scenarios. Next, three scenes involving inter and intra-domain mobility are analysed. They correspond to the first scenario, with the university campus represented by domain B (mobility-enabled WLAN), the cellular network represented by domain C and the user's home network represented by domain A1 (preferred network with no mobility support). The last scene analyses the third scenario, the emergency alert reception. For simplification reasons, in scene 6, the primary notification is received on the WLAN at work and the secondary on the LTE (MBMS), rather than the better suited LTE and DVB-SH access technologies described in Section 3.1. The CCF is actually flexible enough to accommodate as many access technologies as the MT can fit.

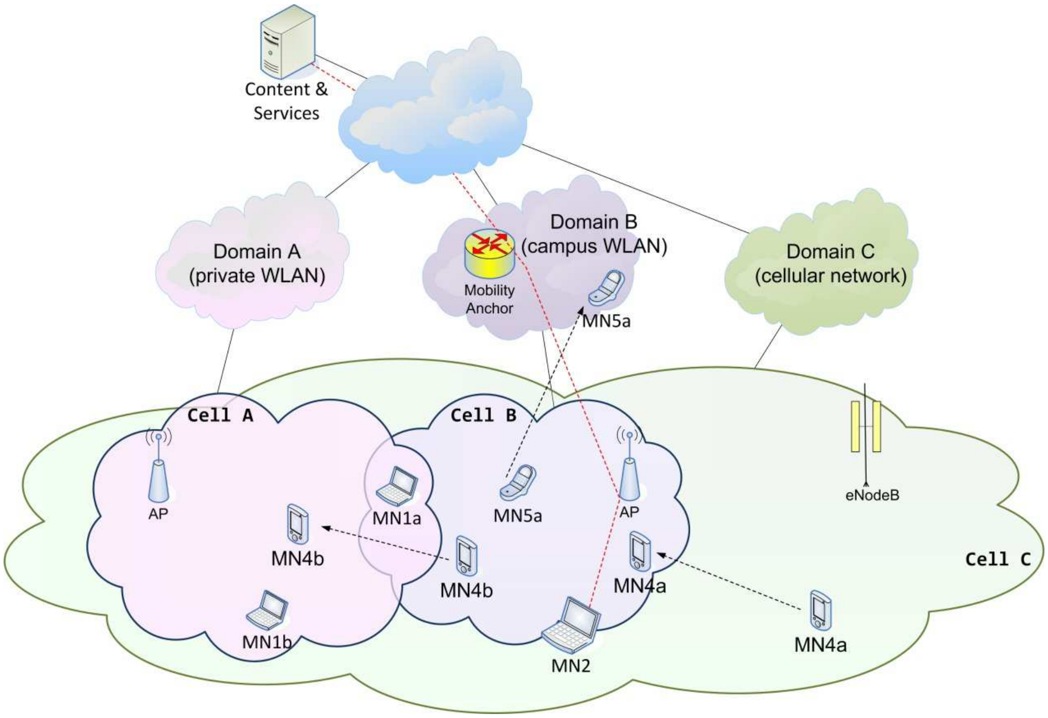


Figure 30: Network layout for the operational analysis

Each generic scene is detailed in a step-by-step manner in the next paragraphs and pictured into flows with the signalling, data and parameters exchanged between the involved components. These scenes highlight the main principles of the framework, the Media Independent Services provided through the MISF, the global coordination performed by the CM to combine the services offered by the GSEs and the cross-layer learning performed by the CLA, in order to obtain a more efficient system. For the global text readability purpose, similar operation descriptions have been repeated in each case, preferably to introducing too many cross-references between steps of the different scenes.

Scene 0 - Mobile Terminal configuration

During configuration, the MT receives and stores in the LIB the knowledge of the following information, either through the user interface or through any other mechanism (e.g., retrieved from the network after an initial connection).

1. User preferences : networks, rate plans and policies;
2. Applications requirements (QoS, network capabilities, multicast, technology...); applications can be identified by their protocols and used port numbers;
3. Network credentials : virtual SIM card for cellular network, L2 or Network credentials for Wi-Fi networks;
4. MT capabilities (technologies, battery, multimedia, other devices such as GNSS or sensors...).

This knowledge serves as a base to be completed by dynamic information learned from the mobile operation and is kept in a persistent manner in the mobile.

Scene 1 - Mobile Terminal is started and attaches the network

Scene 1a - Networks available are A, B, C - A and B are not known - User is authorised on B and not on A:

1. Initialization step: all software components and interfaces are started.
2. The Cognitive Manager (CM) triggers the monitoring of available networks. This task is passed to the Network Access GSE, supported by the MISF and the wireless accesses. They report the network capabilities, Layer 2 (L2) security requirements and status (Q_A,Q_B, Q_C) of networks NetA, NetB and NetC back to the NAGSE. Relevant information is stored in the LIB by the MISF, which acts here as a standard software entity rather than a simple relay.
3. The list of networks found is checked against the LIB knowledge. When available, the parameters, policies of each known network (here only network NetC is known) and related user preferences are returned to the NAGSE.
4. The NAGSE runs its algorithm to select the preferred ordered list of networks. In this case, only one network will be selected and the ordered list of networks is [A, B, C]. This ordered list is returned to the CM for its final decision making.
5. Since networks NetA and NetB were on top of the list and still unknown, CM decides to get support from the mobile user through the UIA. In this case, the user replies that NetA is not OK but NetB is fine. These new user preferences are stored in the LIB to be used later whenever needed, avoiding to ask the mobile user again.
6. The CM triggers the connection to network NetB though the Mobility GSE and the MISF. The WLAN network interface is activated and attached to the network, whose link is not secured at L2 level. A signalling path is used to perform security authentication and mobility registration and binding, since this network hosts a Home Agent. Moreover, the MGSE retrieves the IP address of the interface from the Networking Services to be used as EnvA. The knowledge about this connection is stored in the LIB and a positive feedback sent to the CM.
7. Network NetB is unknown, so the CM triggers a check whether an authentication is required at network level through the Session GSE. A testing http/IP session is opened, which confirms that credentials are required. They are retrieved from the user who completes his authentication, while the credentials are saved in the LIB. At this stage, the initial attachment is terminated.

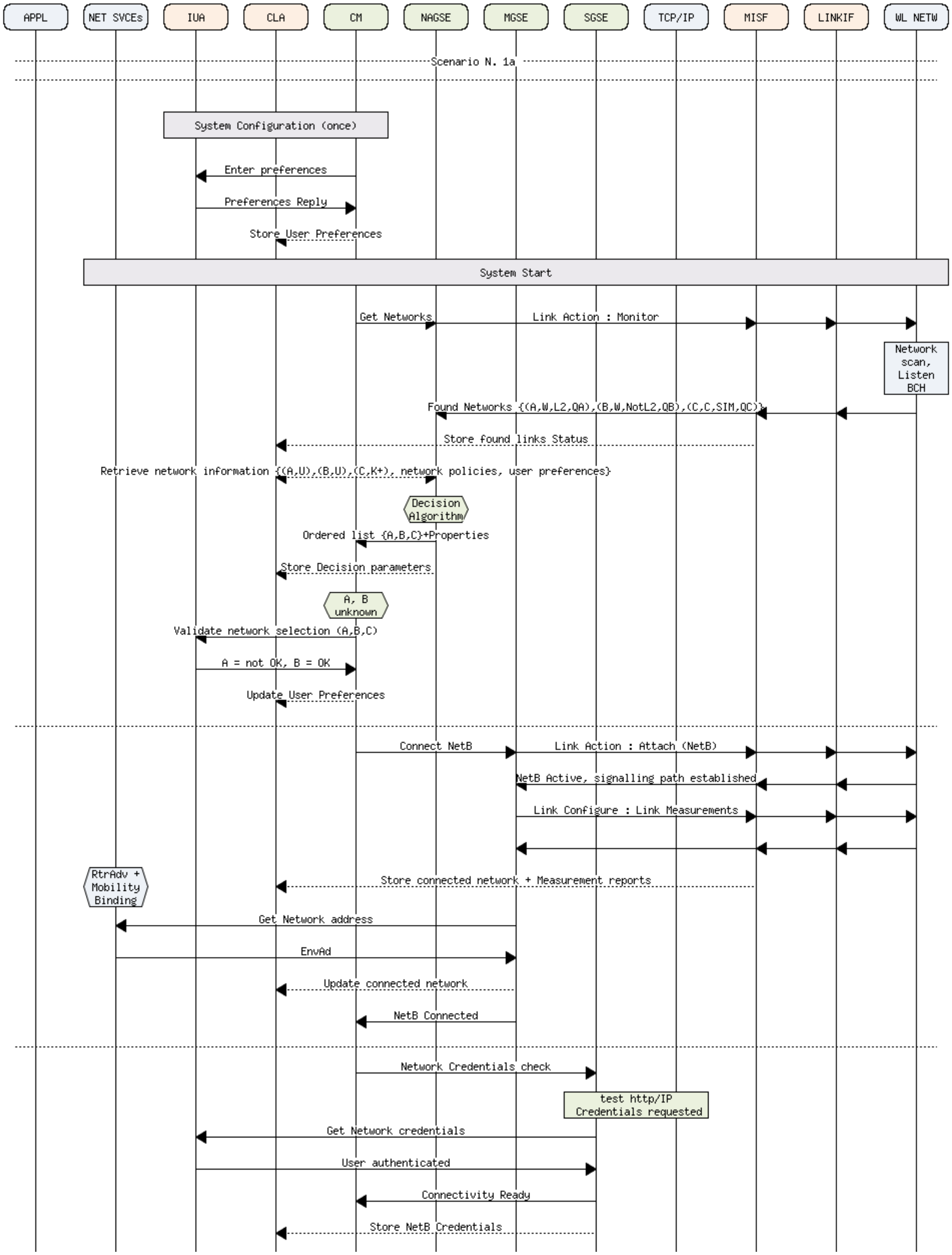

APPL
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 1a
System Configuration (once)
Enter preferences
Preferences Reply
Store User Preferences
System Start
Get Networks
Link Action : Monitor
Network scan, Listen BCH
Found Networks {(A,W,L2,QA),(B,W,NotL2,QB),(C,C,SIM,QC)}
Store found links Status
Retrieve network information {(A,U),(B,U),(C,K+), network policies, user preferences}
Decision Algorithm
Ordered list {A,B,C}+Properties
Store Decision parameters
A, B unknown
Validate network selection (A,B,C)
A = not OK, B = OK
Update User Preferences
Connect NetB
Link Action : Attach (NetB)
NetB Active, signalling path established
Link Configure : Link Measurements
RtrAdv + Mobility Binding
Store connected network + Measurement reports
Get Network address
EnvAd
Update connected network
NetB Connected
Network Credentials check
test http/IP Credentials requested
Get Network credentials
User authenticated
Connectivity Ready
Store NetB Credentials

Figure 31: Scene 1a - Mobile start-up and attachment to network B

Scene 1b – Alternative: Networks available are A and C - A is known but on the black list because the user does not own credentials for A:

1. Initialization step: all software components and interfaces are started.
2. The CM triggers the monitoring of available networks. This task is passed to the NAGSE, supported by the MISF and the wireless accesses. They report the network capabilities, L2 security requirements and status of networks NetA and NetC back to the NAGSE. Relevant information is stored in the LIB by the MISF.
3. The list of networks found is checked against the LIB knowledge. The reply contains the information that NetA is known but on the black list while NetC is known and authorized. NetC parameters, policies and related user preferences are also returned to the NAGSE.
4. The NAGSE runs its algorithm to confirm the suitability of network NetC. In this case, the ordered list of networks is limited to [C]. This ordered list is returned to the CM for its final decision making.
5. Since network NetC is known, there is no need to confirm this decision with the user. The CM triggers the connection to network NetC though the MGSE and the MISF. A signalling path in the cellular network is activated and the MGSE retrieves the IP address of the interface bound to this network from the NS, to be used as EnvA. The knowledge about this connection is stored in the LIB and a positive feedback sent to the CM. At this stage, the initial attachment is terminated.

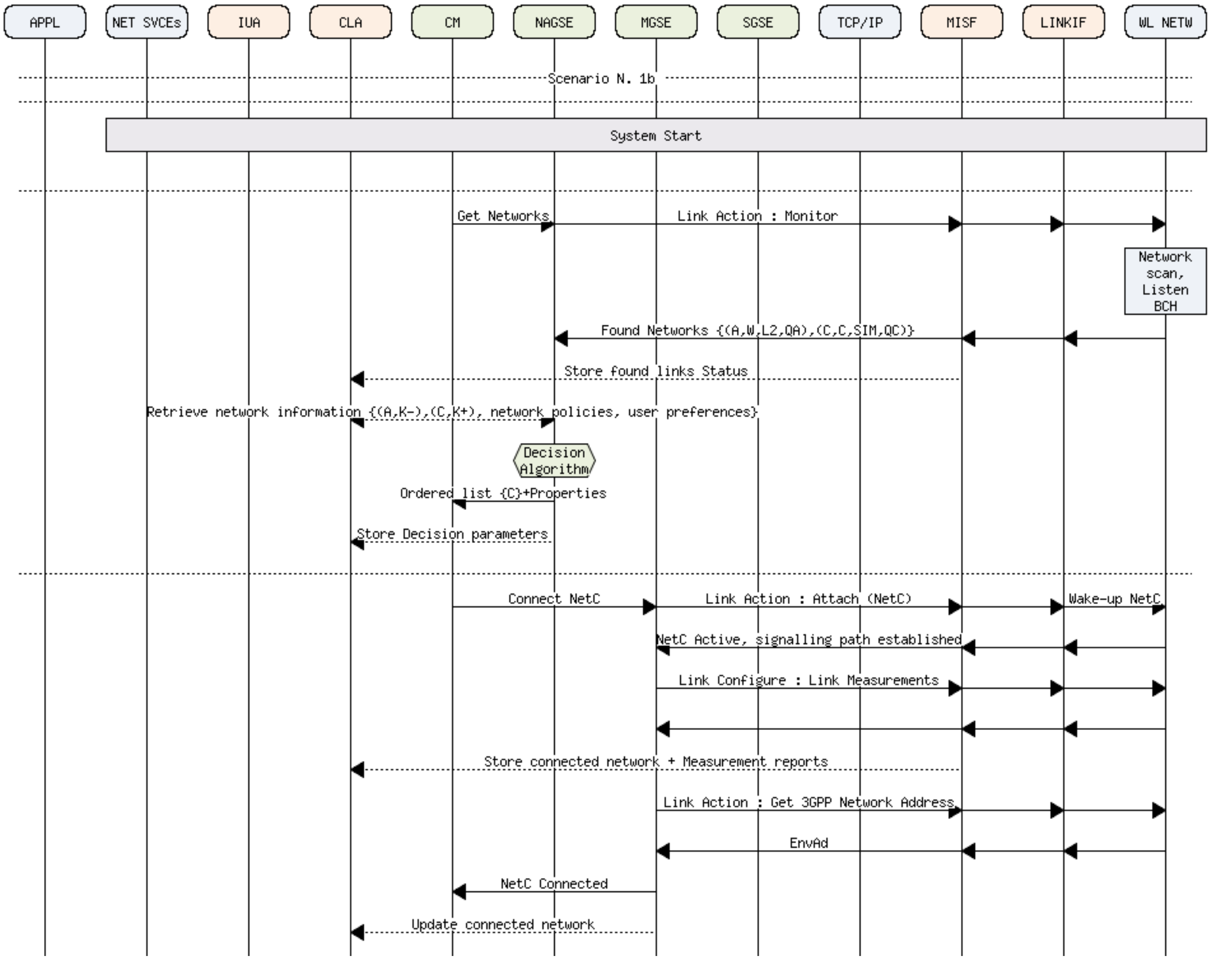

APPL
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 1b
System Start
Get Networks
Link Action : Monitor
Network scan, Listen BCH
Found Networks {(A,W,L2,QA),(C,C,SIM,QC)}
Store found links Status
Retrieve network information {(A,K-),(C,K+), network policies, user preferences}
Decision Algorithm
Ordered list {C}+Properties
Store Decision parameters
Connect NetC
Link Action : Attach (NetC)
Wake-up NetC
NetC Active, signalling path established
Link Configure : Link Measurements
Store connected network + Measurement reports
Link Action : Get 3GPP Network Address
EnvAd
NetC Connected
Update connected network

Figure 32: Scene 1b - Alternative: Mobile start-up and attachment to network C

Scene 2 - Session setup – An application starts

1. When an application starts, it opens a socket on the Virtual Socket interface to the CM, which then retrieves the application requirements from the LIB.
2. The CM triggers the NAGSE to check that the current connected network is suitable for this new application. The decision algorithm is run and here, it confirms to the CM that the network is fine.
3. The CM triggers the session establishment from the SGSE on this network, providing the application QoS requirements. Based on the application Traffic Flow Template (TFT, indicates QoS and bit rate), the necessary resources are requested to this network via the MISF, checking afterwards that the QoS requirements are fulfilled.
4. The SGSE is ready to open a real socket to the TCP/IP protocol stack and allocates a PA to the application. The PA is bound to the current network address EnvA. The knowledge about this session, including the mapping between the two addresses, is stored in the LIB and a positive feedback is sent to the CM and the application. The data transfer path in the mobile is ready, the application can start.

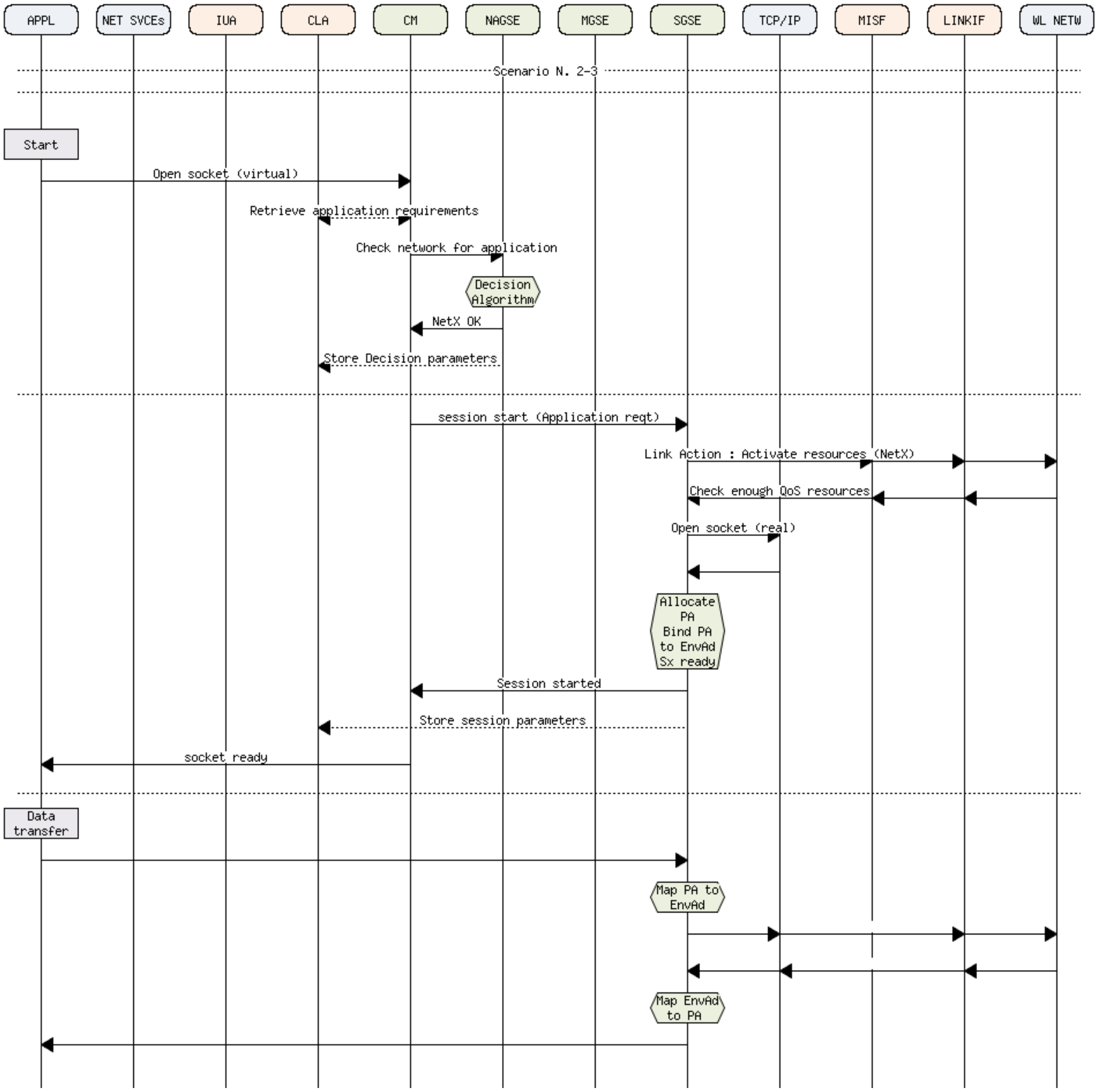


Figure 33: Scene 2 and 3 - Session setup, application starts and data transfer

Scene 3 - Data transfer

Within the CCF, the data packets flow through the following path:

1. From the application to the SGSE, packets are sent through the virtual socket interface, with the following addresses binding:
    - Local address= PA ; remote address = CN address
2. The SGSE translates the PA to EnvA and sends packets to the transport protocol through the real socket interface, with the following addresses binding:
    - Local address = EnvA ; remote address = CN address
3. The packet is transferred by the TCP/IP stack to the Link Interface which forwards it on the Link wireless access (LLC).

Scene 4 - Inter-domain mobility

Based on the first scenario described in Section 3.1, the set of possible inter-domain mobility use cases varies according to three dimensions, according to the properties of the target network {known/unknown; provides mobility support or not; the handover is either soft (the old connection is broken after the handover completion) or hard (the old connection is lost before the handover starts)}. This means that from a technical perspective, nine cases should be considered. To avoid too much repetition, only the two extreme cases are described below. The first one corresponds to the triplet {network known; no mobility support; soft handover}. The second one shows the operation for the triplet {network unknown; mobility support in the target network; hard handover}. In all cases, the user is supposed to own the necessary credentials to be able to access the target network.

Scene 4a - Mobility from C to A1 (user's home) - A1 is known - soft handover

Preparation

1. The mobile is connected to network NetC, running an application which uses TCP as transport protocol.
2. An event is raised at the MISF by a local sensor, and forwarded to the CM through the MGSE, that the mobile user is arriving at his home, which is a well-known configured location.
3. The CM retrieves the application requirements from the LIB and triggers the NAGSE to check whether this network NetA1 is better suited for the currently running application. The decision algorithm is executed and here, it confirms to the CM that the network NetA1 is preferred.

Execution

4. The CM retrieves the session parameters from the LIB. In this case, the transport protocol used is TCP, so changing the network from NetC to NetA1 will mean breaking the connection. The CM triggers the SGSE to prepare the session break and to buffer the data packets with the light DTN mechanism for the handover duration.
5. The CM triggers the connection to network NetA1 though the MGSE and the MISF. The WLAN network interface is activated. NetA1 link is secured at L2 level, so the L2 credentials are retrieved from the LIB by the MISF which requests the attachment to the PoA. When it receives a successful status, the MGSE retrieves the IP address of the interface from the NS to be used as EnvA. The knowledge about this connection is stored in the LIB and a positive feedback sent to the CM.
6. The CM commands the SGSE to transfer the current session to network NetA1, providing the application QoS requirements. Based on the application TFT (QoS and bit rate), the necessary QoS should be requested to this network via the MISF, but this step is skipped here (network NetA1) because this usually does not apply to WLAN accesses.

7. The SGSE is ready to open a real socket to the TCP/IP protocol stack. The session PA is bound to the new network EnvA address. The data path is switched to this new binding, buffered packets are forwarded on the new path and a positive feedback is sent to the CM and the application. The knowledge about this session, including the mapping between the two addresses, is stored in the LIB.
8. The application data transfer can resume normally.

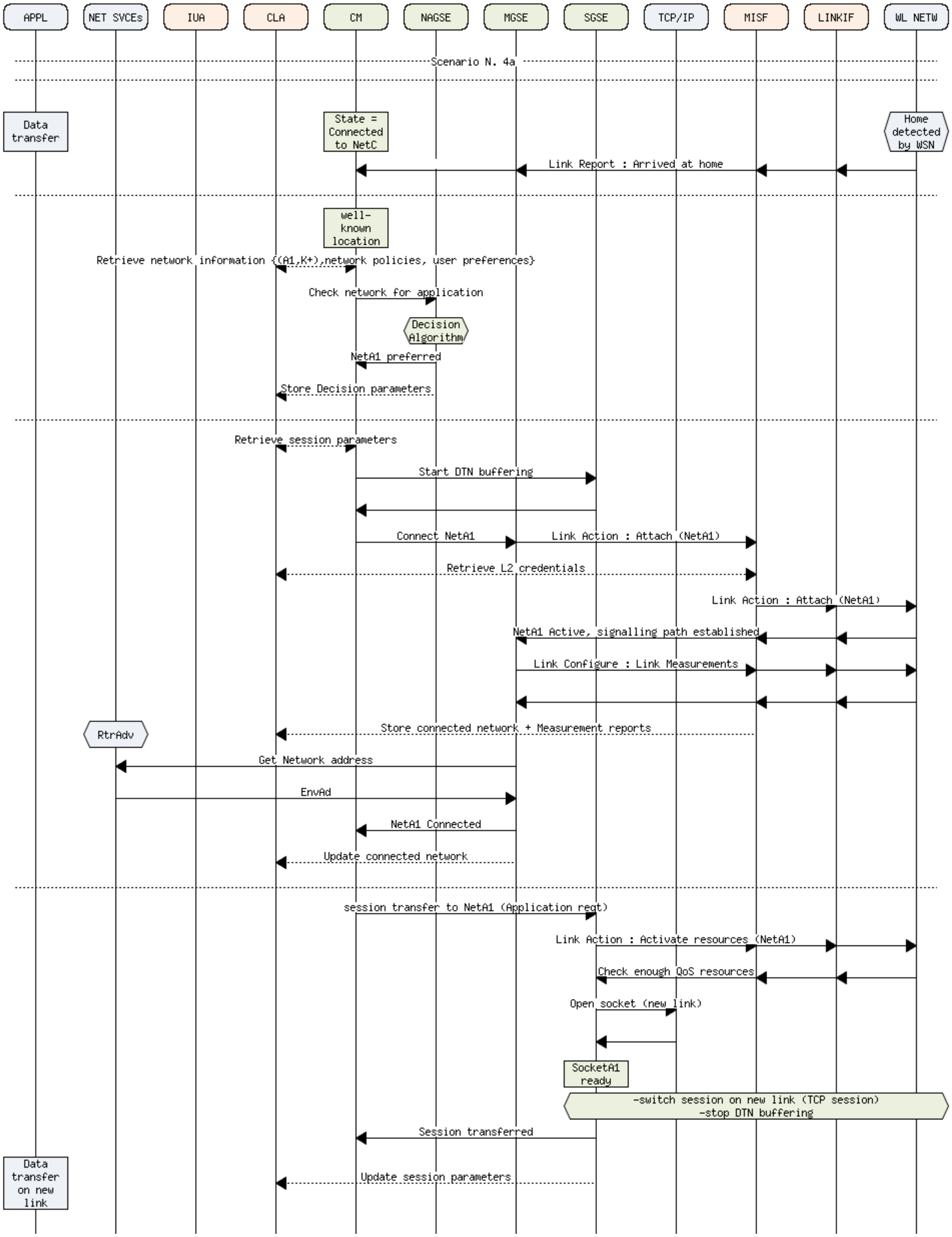

APPL
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 4a
Data transfer
State = Connected to NetC
Home detected by WSN
Link Report : Arrived at home
well-known location
Retrieve network information {(A1,K+),network policies, user preferences}
Check network for application
Decision Algorithm
NetA1 preferred
Store Decision parameters
Retrieve session parameters
Start DTN buffering
Connect NetA1
Link Action : Attach (NetA1)
Retrieve L2 credentials
Link Action : Attach (NetA1)
NetA1 Active, signalling path established
Link Configure : Link Measurements
RtrAdv
Store connected network + Measurement reports
Get Network address
EnvAd
NetA1 Connected
Update connected network
session transfer to NetA1 (Application reqt)
Link Action : Activate resources (NetA1)
Check enough QoS resources
Open socket (new link)
SocketA1 ready
-switch session on new link (TCP session)
-stop DTN buffering
Session transferred
Data transfer on new link
Update session parameters

Figure 34: Scene 4a - Inter-domain mobility from C to A1 (user's home) with soft handover

Scene 4b - Mobility from A1 to B - B is unknown - hard handover

Preparation

1. The mobile is connected to network NetA1, running an application which uses TCP as transport protocol.
2. An event is raised at the MISF and forwarded to the CM through the MGSE. It reports that the quality of network NetA1 is rapidly decreasing.
3. The CM retrieves the application requirements and the session parameters from the LIB. In this case, the transport protocol used is TCP, so changing the network from NetA1 to NetB without the support of a mobility mechanism, will mean breaking the session. The CM immediately triggers the SGSE to prepare the session break and to buffer the data packets with the light DTN mechanism for the handover duration.
4. The CM then triggers the monitoring of available networks. This task passed to the NAGSE, supported by the MISF and the wireless accesses. They report the network capabilities, L2 security requirements and status of networks NetB and NetC back to the NAGSE. Relevant information is stored in the LIB by the MISF.
5. The list of networks found is checked against the LIB knowledge. When available, the parameters, policies of each known network (here only network NetC is known) and related user preferences are returned to the NAGSE.
6. The NAGSE runs its algorithm to select the preferred ordered list of networks. In this case, only one network will be used by the application and the ordered list of networks is [B, C]. This ordered list is returned to the CM for its final decision making.
7. Since network NetB is on top of the list and yet unknown, CM decides to get support from the mobile user through the UIA. In this case, the UIA replies that NetB is OK. These new user preferences are stored in the LIB to be used later whenever needed, avoiding to ask the mobile user again.

Execution

8. The CM triggers the connection to network NetB though the MGSE and the MISF. They activate the switching to the NetB network, whose link is not secured at L2 level and request the attachment to the new PoA. A signalling path is used to perform security authentication and mobility registration and binding, since this network hosts a HA. Next, the MGSE retrieves the IP address of the interface from the NS to be used as EnvA. The knowledge about this connection is stored in the LIB and a positive feedback sent to the CM.
9. The CM commands the SGSE to transfer the current session to network NetB, providing the application requirements for QoS. Based on the application TFT (QoS and bit rate), the necessary QoS should be requested to this network via the MISF, but this step is skipped here because this usually does not apply to WLAN accesses.
10. Network NetB is unknown, so the CM triggers a check whether an authentication is required at network level through the SGSE. A testing HTTP/IP session is opened, which confirms that credentials are required. They are retrieved from the user who completes his authentication, while the credentials are saved in the LIB.
11. The SGSE is ready to open a real socket for the application to the TCP/IP protocol stack. The session PA is bound to the new network EnvA address. The data path is switched to this new binding, buffered packets are forwarded on the new path, buffering is stopped and a positive feedback is sent to the CM and the application. The knowledge about this session, including the mapping between the two addresses, is stored in the LIB.
12. The application data transfer can resume normally.

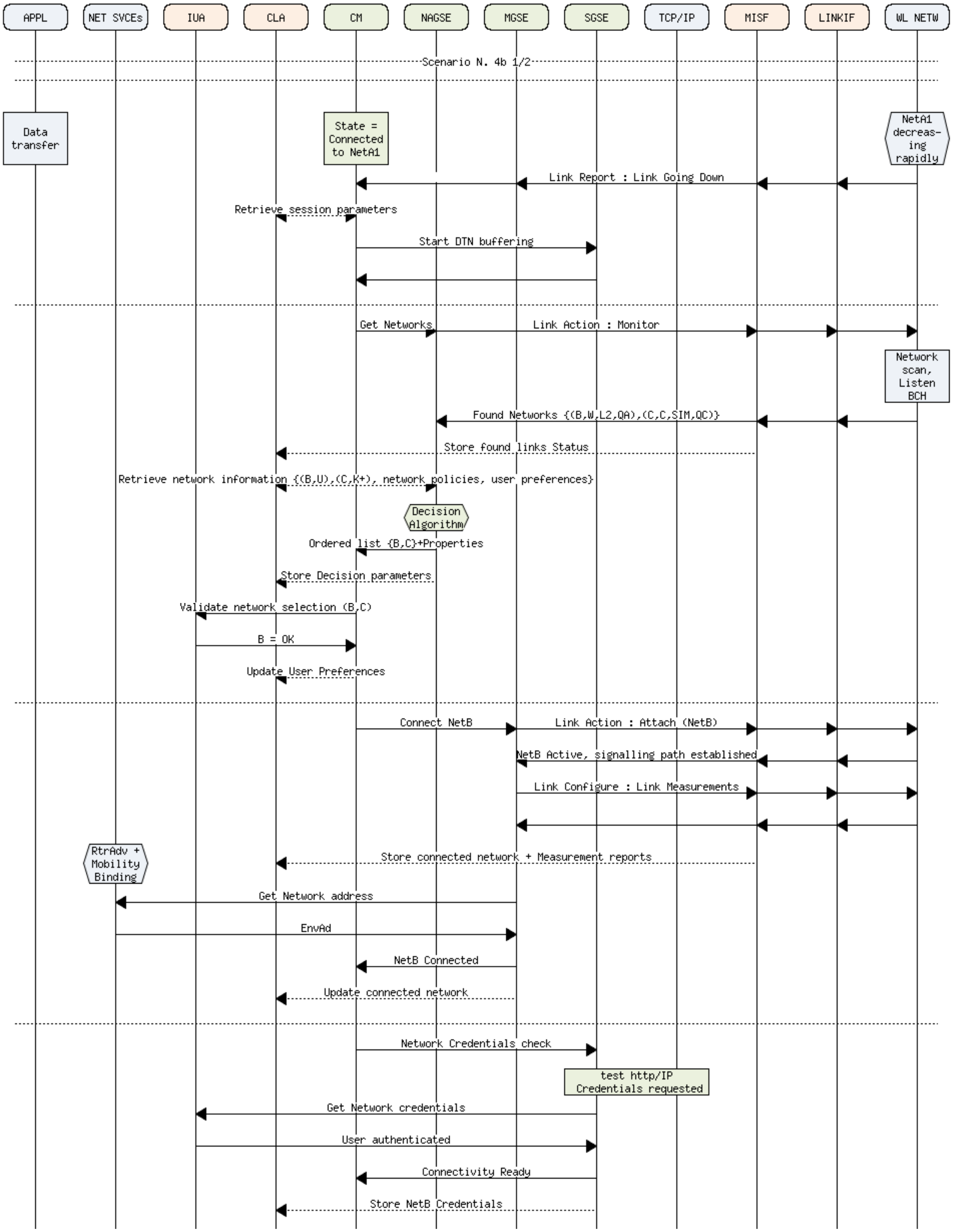

APPL
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 4b 1/2
Data transfer
State = Connected to NetA1
NetA1 decreas-ing rapidly
Link Report : Link Going Down
Retrieve session parameters
Start DTN buffering
Get Networks
Link Action : Monitor
Network scan, Listen BCH
Found Networks {(B,W,L2,QA),(C,C,SIM,QC)}
Store found links Status
Retrieve network information {(B,U),(C,K+), network policies, user preferences}
Decision Algorithm
Ordered list {B,C}+Properties
Store Decision parameters
Validate network selection (B,C)
B = OK
Update User Preferences
Connect NetB
Link Action : Attach (NetB)
NetB Active, signalling path established
Link Configure : Link Measurements
RtrAdv + Mobility Binding
Store connected network + Measurement reports
Get Network address
EnvAd
NetB Connected
Update connected network
Network Credentials check
test http/IP Credentials requested
Get Network credentials
User authenticated
Connectivity Ready
Store NetB Credentials

Figure 35: Scene 4b Part 1/2 - Inter-domain mobility from A1 to B with hard handover

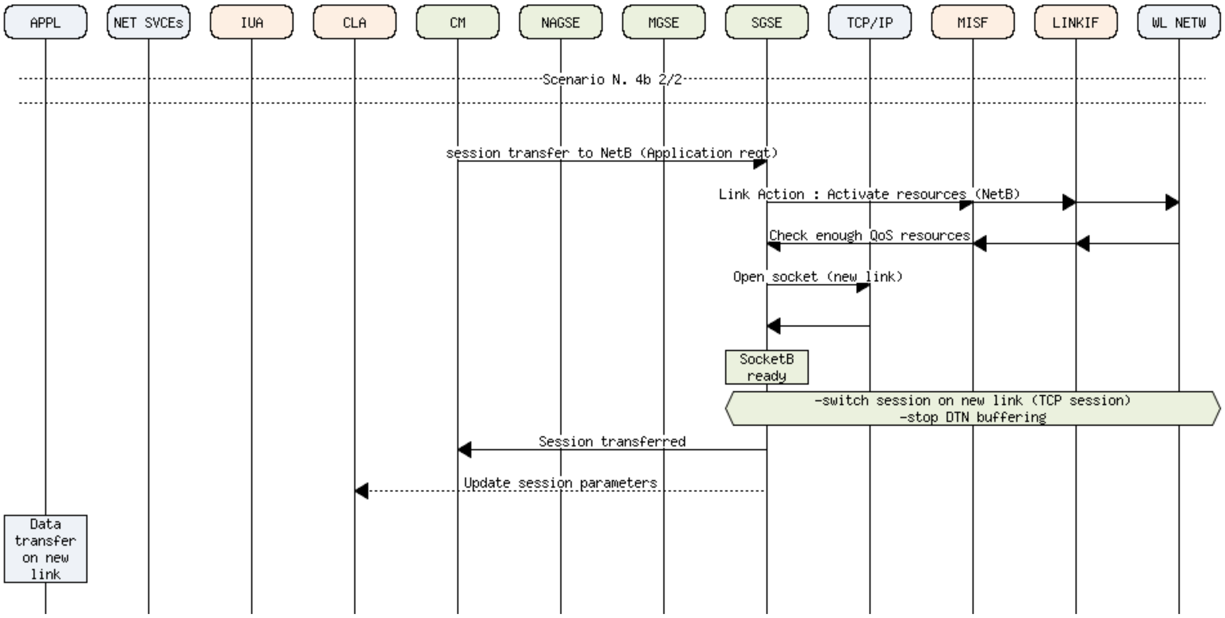

APPL
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 4b 2/2
session transfer to NetB (Application reqt)
Link Action : Activate resources (NetB)
Check enough QoS resources
Open socket (new link)
SocketB ready
-switch session on new link (TCP session)
-stop DTN buffering
Session transferred
Update session parameters
Data transfer on new link

Figure 36: Scene 4b Part 2/2 - Inter-domain mobility from A1 to B with hard handover

Scene 5 - Intra-domain mobility

Scene 5a - Mobility inside B - soft handover (B is mobility-enabled)

Preparation

1. The mobile is connected to network NetB, running an application which uses TCP as transport protocol.
2. An event is raised at the MISF and forwarded to the CM through the MGSE. It reports that the quality of network NetB is rapidly decreasing.
3. The CM retrieves the application requirements and the session parameters from the LIB. In this case, the transport protocol used is TCP, so changing the PoA on the same technology will mean a short break of the connectivity. The CM triggers the SGSE to prepare the break and buffer the data packets with the light DTN mechanism for the switching duration.
4. The CM triggers the monitoring of available networks. This task is put in charge of the NAGSE, supported by the MISF and the wireless accesses. They report the network capabilities, L2 security requirements and status of networks NetB with a new PoA and NetC back to the NAGSE. Relevant information is stored in the LIB by the MISF.
5. The list of networks found is checked against the LIB knowledge. When available, the parameters, policies of each known network (here both networks are known) and associated user preferences are returned to the NAGSE.
6. The NAGSE runs its algorithm to select the preferred ordered list of networks. In this case, only one network will be used by the application and the ordered list of networks is [B, C]. This ordered list is returned to the CM for its final decision making.
7. Since network NetB has been used previously, there is no need to confirm this decision with the user.

.

<u>Execution</u>

8. The CM triggers the connection to the new PoA of network NetB though the MGSE and the MISF. As this network is deployed with MIP and hosts a HA in its domain, a signalling path is used to perform the new mobility binding. Next, the MGSE retrieves the new CoA address of the interface from the NS to be used as EnvA. The knowledge about this connection is stored in the LIB and a positive feedback sent to the CM.
9. No real session transfer is triggered since NS handle the session continuity, but on CM request, packets buffered in the SGSE are forwarded to the new path and a positive feedback is sent to the CM. The knowledge about this session, including the mapping between the two addresses, is stored in the LIB.
10. The application data transfer can resume normally.

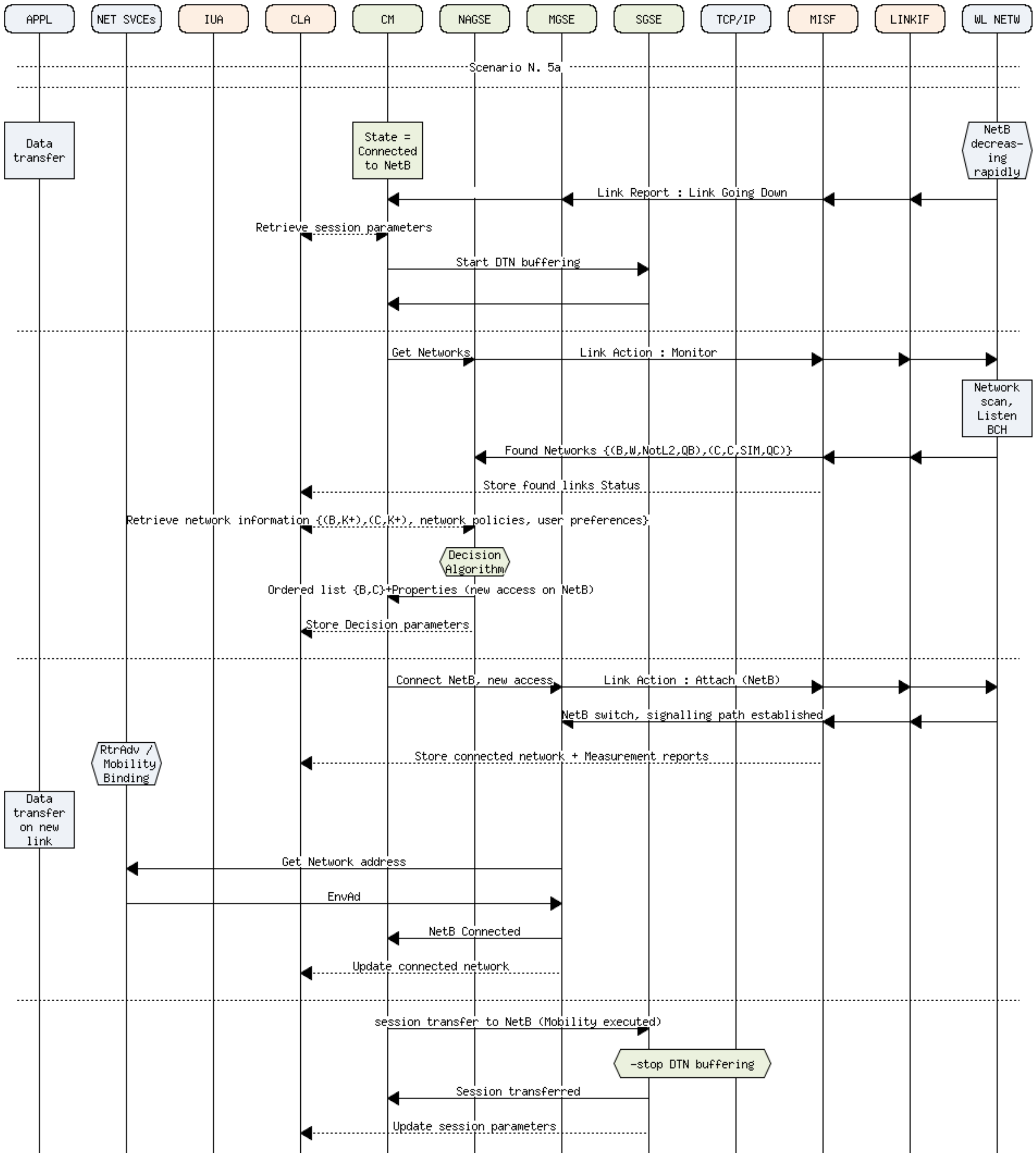

APPL
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 5a
Data transfer
State = Connected to NetB
NetB decreas-ing rapidly
Link Report : Link Going Down
Retrieve session parameters
Start DTN buffering
Get Networks
Link Action : Monitor
Network scan, Listen BCH
Found Networks {(B,W,NotL2,QB),(C,C,SIM,QC)}
Store found links Status
Retrieve network information {(B,K+),(C,K+), network policies, user preferences}
Decision Algorithm
Ordered list {B,C}+Properties (new access on NetB)
Store Decision parameters
Connect NetB, new access
Link Action : Attach (NetB)
NetB switch, signalling path established
RtrAdv / Mobility Binding
Store connected network + Measurement reports
Data transfer on new link
Get Network address
EnvAd
NetB Connected
Update connected network
session transfer to NetB (Mobility executed)
-stop DTN buffering
Session transferred
Update session parameters

Figure 37: Scene 5a – Intra-domain mobility using mobility mechanism in network B

Scene 5b - Mobility inside C

This operation is completely transparent to the CCF framework. There is no network change, no IP address change; everything is performed internally in the cellular network using 3GPP mobility mechanisms, as described in Section 2.3.

Scene 6 – Multi-homing

In the case of an emergency alert, an application 1 running on network A starts another application 2 which requires MBMS on network C. After a while, the connection with network A is lost and the application 1 is transferred to network C as well.

Multi-homing for a second application

1. The mobile is connected to network NetA, running the Application1 which uses TCP as transport protocol.
2. Application1 triggers the launch of Application2 which opens a socket on the Virtual Socket interface to the CM. First, the CM retrieves the Application2 requirements from the LIB.
3. The CM triggers the NAGSE to check that the network NetA is suitable for the new application. The decision algorithm is executed and here, it confirms to the CM that network NetA is not fine.
4. The CM triggers the monitoring of available networks. This task is put in charge of the NAGSE, supported by the MISF and the wireless accesses. They report the network capabilities, L2 security requirements and status of networks NetA and NetC back to the NAGSE. Relevant information is stored in the LIB by the MISF.
5. The list of networks found is checked against the LIB knowledge, eliminating NetA due to the previous algorithm result. The reply shows that NetC is a known network. Its parameters, policies and associated user preferences are returned to the NAGSE.
6. The NAGSE runs its algorithm to confirm the preferred ordered list of networks suitable to Application 2. In this case, only network NetC can be used since MBMS is requested and the ordered list of networks is [C]. This list is returned to the CM for its final decision making.
7. Since network NetC is known, there is no need to confirm this decision with the user. The CM triggers the connection to network NetC though the MGSE and the MISF. A signalling path in the cellular network is activated and the MGSE retrieves the IP address of the interface bound to this network from the NS, to be used as EnvA. The knowledge about this connection is stored in the LIB and a positive feedback sent to the CM.
8. The CM triggers the session establishment from the SGSE on network NetC, providing the application requirements for QoS. Based on the application TFT (QoS and bit rate), the necessary resources are allocated on this network via the MISF, checking afterwards that the QoS requirements are fulfilled.
9. The SGSE is ready to open a real socket to the TCP/IP protocol stack and allocate a PA to the new application. The PA is bound to the EnvA address in NetC network. The knowledge about this session, including the mapping between the two addresses, is stored in the LIB and a positive feedback is sent to the CM and the application. The data transfer path in the mobile is ready, the application can start.

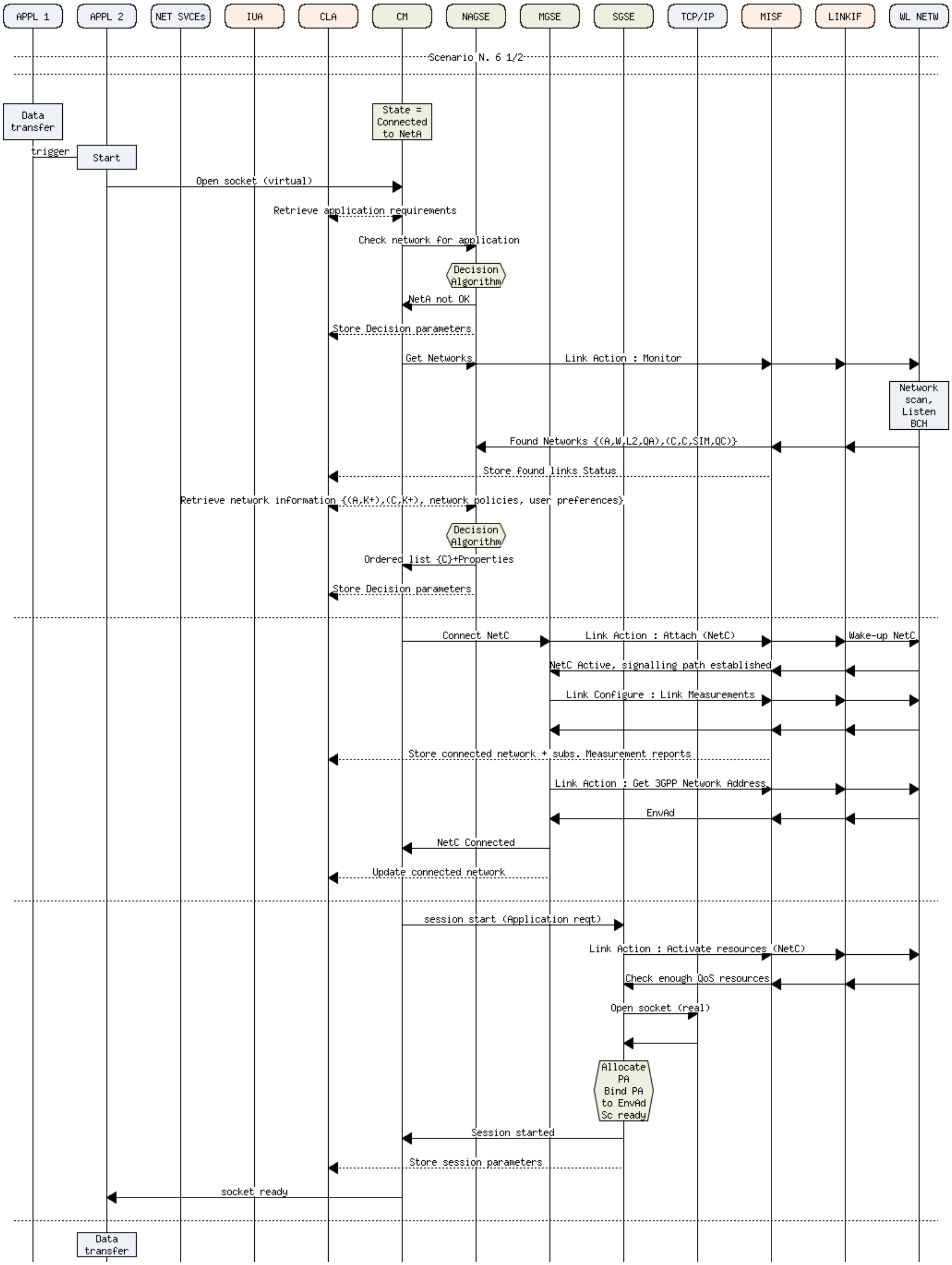

APPL 1
APPL 2
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 6 1/2
Data transfer
State = Connected to NetA
trigger
Start
Open socket (virtual)
Retrieve application requirements
Check network for application
Decision Algorithm
NetA not OK
Store Decision parameters
Get Networks
Link Action : Monitor
Network scan, Listen BCH
Found Networks {(A,W,L2,QA),(C,C,SIM,QC)}
Store found links Status
Retrieve network information {(A,K+),(C,K+), network policies, user preferences}
Decision Algorithm
Ordered list {C}+Properties
Store Decision parameters
Connect NetC
Link Action : Attach (NetC)
Wake-up NetC
NetC Active, signalling path established
Link Configure : Link Measurements
Store connected network + subs. Measurement reports
Link Action : Get 3GPP Network Address
EnvAd
NetC Connected
Update connected network
session start (Application reqt)
Link Action : Activate resources (NetC)
Check enough QoS resources
Open socket (real)
Allocate PA Bind PA to EnvAd Sc ready
Session started
Store session parameters
socket ready
Data transfer

Figure 38: Scene 6 Part 1/2 – Multi-homing with one application per network

Handover

10. An event is raised at the MISF, and forwarded to the CM through the MGSE. It reports that the quality of network NetA is rapidly decreasing.
11. The CM retrieves the applications requirements and their session parameters from the LIB. It appears that Application1 has to be transferred while Application2 can be left unchanged. In this case, the transport protocol used is TCP, so changing the network will mean breaking the connection. The CM triggers the SGSE to prepare the session break and buffer the data packets with the DTN mechanism for the handover duration.
12. The CM triggers the monitoring of available networks. This task is passed to the NAGSE, supported by the MISF and the wireless accesses. They report the network capabilities, L2 security requirements and status of network NetC back to the NAGSE. No network other than the failing NetA is detected on the WLAN interface. Relevant information is stored in the LIB by the MISF.
13. The parameters, policies and associated user preferences of network NetC are retrieved by the NAGSE from the LIB. It runs its algorithm to confirm the preferred ordered list of networks. In this case, only network NetC can be used and the ordered list of networks is [C]. This ordered list is returned to the CM for its final decision making.
14. NetC is accepted by the CM and since it is already active, the CM can proceed directly to the session transfer.
15. The CM requests the SGSE to transfer the current session to network NetC, providing the application requirements for QoS. Based on the application TFT (QoS and bit rate), the necessary resources are allocated in the network via the MISF, checking afterwards that the QoS requirements are fulfilled.
16. The SGSE is ready to open a new socket to the TCP/IP protocol stack. The session PA of Application1 is bound to the EnvA address of network NetC. The data path is switched to this new binding, buffered packets are forwarded on the new path, buffering is stopped and a positive feedback is sent to the CM. The knowledge about this session, including the mapping between the two addresses, is stored in the LIB.
17. Application1 can resume its data transfer.

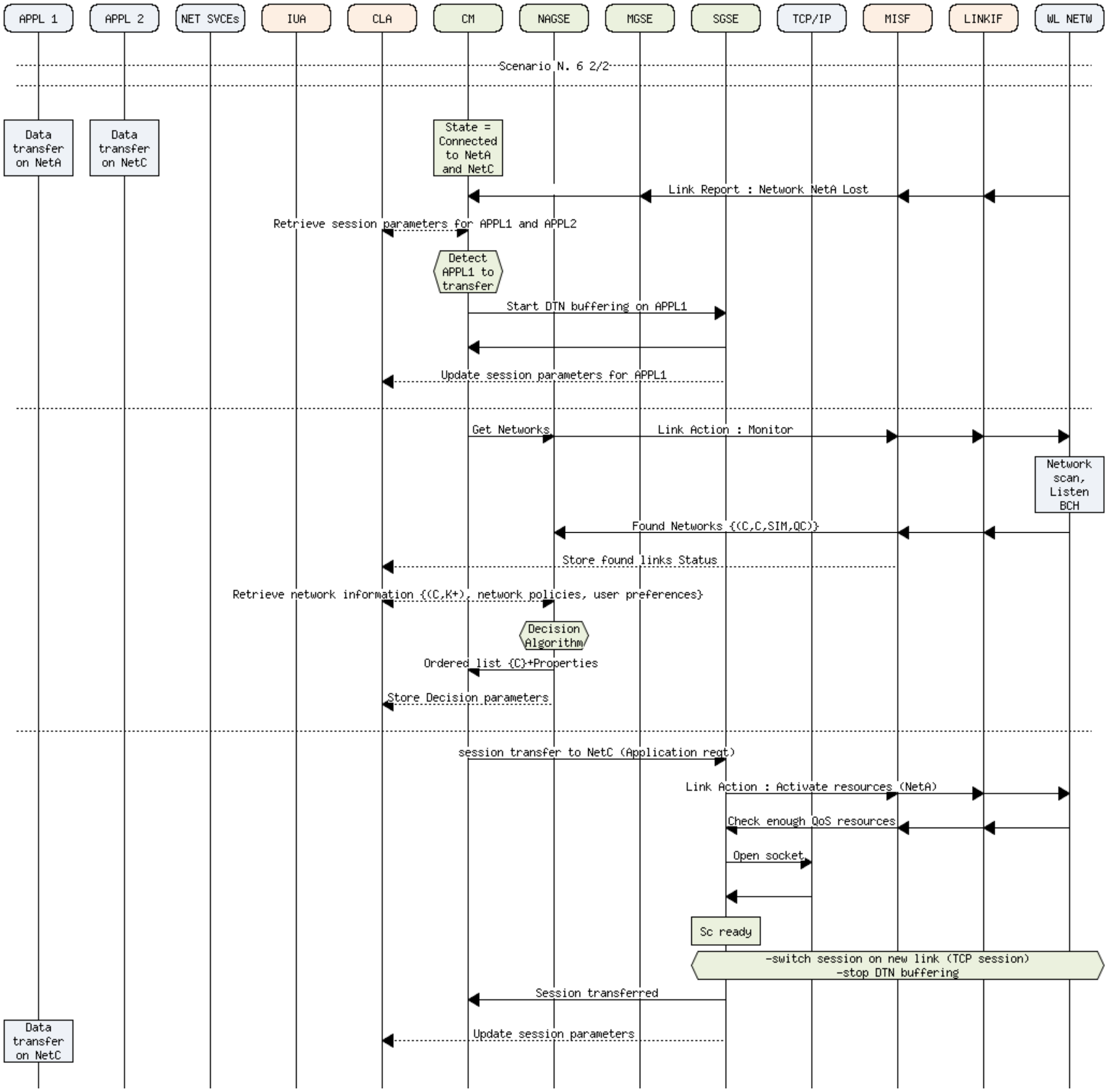

APPL 1
APPL 2
NET SVCEs
IUA
CLA
CM
NAGSE
MGSE
SGSE
TCP/IP
MISF
LINKIF
WL NETW
Scenario N. 6 2/2
Data transfer on NetA
Data transfer on NetC
State = Connected to NetA and NetC
Link Report : Network NetA Lost
Retrieve session parameters for APPL1 and APPL2
Detect APPL1 to transfer
Start DTN buffering on APPL1
Update session parameters for APPL1
Get Networks
Link Action : Monitor
Network scan, Listen BCH
Found Networks {(C,C,SIM,QC)}
Store found links Status
Retrieve network information {(C,K+), network policies, user preferences}
Decision Algorithm
Ordered list {C}+Properties
Store Decision parameters
session transfer to NetC (Application reqt)
Link Action : Activate resources (NetA)
Check enough QoS resources
Open socket
Sc ready
-switch session on new link (TCP session)
-stop DTN buffering
Session transferred
Data transfer on NetC
Update session parameters

Figure 39: Scene 6 Part 2/2 – Multi-homing with one application per network

Summary

In this section, the execution of a few generic scenes extracted from the application typical scenarios presented in Section 3.1 has been explained. They cover the main networking functionalities necessary to handle the terminal operation that have been described at the beginning of this chapter, from terminal initialization, application start-up to inter- and intra-domain mobility. The last scene introduces a more complex roaming situation, where two applications are started and run in parallel on two different access networks. These scenes have shown how the basic building blocks proposed in the CCF can accommodate a varied set of events occurring in a mobile terminal when it roams across heterogeneous networks. The main principles of the CCF, i.e. technology abstraction, orchestration of generic service enablers and shared knowledge, cooperate to provide a simple yet flexible framework, able to enhance the efficiency of the terminal connectivity and avoid the failure cases described in Section 2.1.

4.2 Interactions Between Components

Based on the generic scenes and flows presented in the previous section, the interactions between the components of the CCF and externally with their environment could be defined. The first part of this section identifies the services that are provided to the upper layer entities other than the applications, looking now similar to the Open Connection Manager API architecture that has been published very recently by the OMA (see Section 2.4). These entities are reunited under a single “Networking Services” block. The MT has to support interactions with the remote network nodes. The functions required by the various entities in the network are associated with the components of the framework. This is described in the second part. Finally, the interfaces, whether internal or external to the CCF, are specified at high level in a third part. A detailed specification of the internal interfaces, including the exchanged parameters, has been proposed in ANNEX A to complete the system description.

4.2.1 Services Provided to Upper Layer Components

The services provided by the CCF to the legacy upper layer entities in the control plane are pictured as Networking Services on top right of Figure 29. The main service in relation with this study is the *connectivity and handover service*. It supports the establishment or transfer of the network connectivity to the best available network at a given time, sharing the traffic between one or several links for better performance and load balancing when required and possible. This service is provided through an interface that extends the User MIH_SAP described in Section 2.2. Secondly, the CCF keeps an exhaustive set of *statistics and internal information* about the MT and is able to provide it on request. These data include, but are not restricted to, the network address, the data traffic statistics, the current geographical location and the reference time retrieved from a GNSS or sensor device. A more complete list of these data is defined with the LIB database, in Section 4.3.1. Thirdly, *Power management* can be assisted by the framework. Through the CLA, the CCF would also be flexible enough to support a mobile terminal *configuration service* or possibly some *AAA and security control function*. The two latter services have not been developed further in this thesis because they are not essential to solve the connectivity problem, although they perfectly fit in the framework. Finally, the CCF design allows implementing the very recent Open Connection Manager API described in Section 2.4. It would be located above the CA, inserting, when necessary, additional GSEs to comply with the OMA architecture.

4.2.2 Interactions with the Network

Part of the mobile operations involve interactions with the network, and thus heavily rely on the type of the network and how it is configured. They are handled by the Networking Services, the IP protocol stack and the Wireless Access components. These interactions are not affected by the CCF framework, but may be assisted by the provided services listed in Section 4.2.1. The following list indicates which component of the framework in the MT, as shown in Figure 29, is responsible for interacting with the network.

- Network discovery and monitoring: Wireless Access managed by the NAGSE;
- Link establishment and release: Wireless Access managed by the MGSE;
- Network security, including access control, registration and authentication and billing support : AAA Networking Service, assisted by the CCF;
- IP address allocation or generation (prefix retrieval) : IP protocol stack;
- QoS resource negotiation (mainly for cellular networks, which are connection-oriented; no negotiation in WLAN networks which are contention based): Wireless Access managed by the SGSE;
- Mobility protocols when available, e.g., HA discovery and binding: mobility management Networking Service, assisted by the CCF;
- Retrieval of configuration parameters, software updates, security credentials and policies from the network: configuration management Networking Service, assisted by the CLA.

4.2.3 Global Interfaces Definition

Figure 40 takes up Figure 29 presented in Section 4.1.2, highlighting the various interfaces which can be encountered in the CCF.

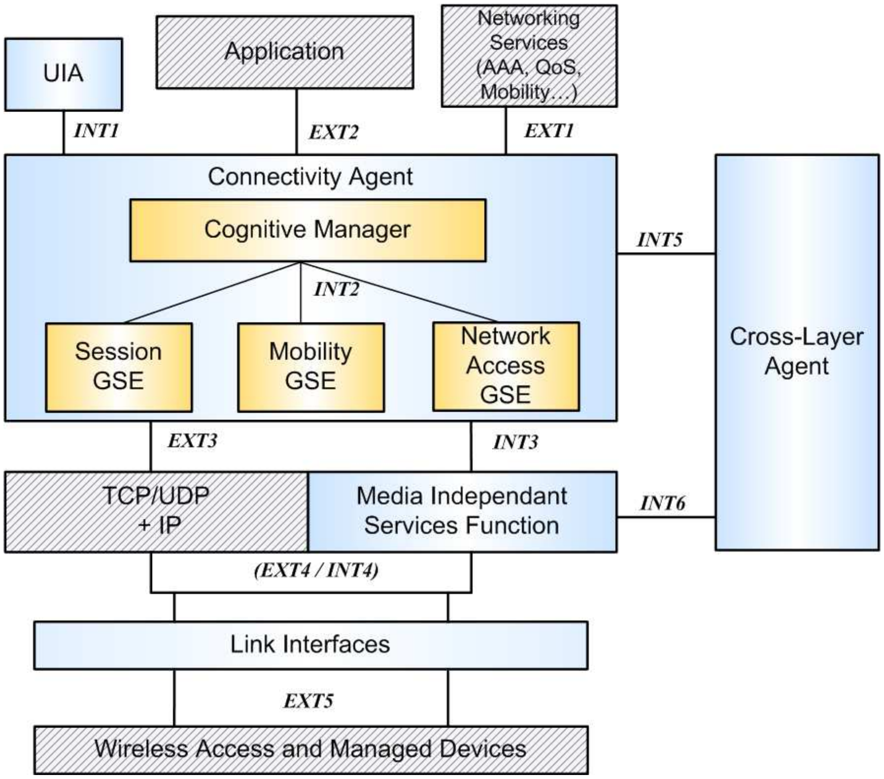


Figure 40: Interfaces of the CCF

4.2.3.1 External interactions

Most of the external interfaces are unchanged and comply with existing APIs, except for the User MIH_SAP included in interface *EXT1* which is enhanced thanks to the larger functionality range of the generic services.

Below is the list of main interfaces with components external to the CCF.

EXT1: Networking Services ←→ Connectivity Agent

This interface operates on the control plane. It provides to the upper layers the services described in Section 4.2.1. It includes, but is not limited to, a User MIH_SAP.

EXT2: Legacy Application ←→ Connectivity Agent

This interface operates on the data plane. It is a virtual socket interface; it behaves the same as a standard Berkeley socket API from the application point of view. It provides the following operations:

- Open, bind and close a socket on the Virtual Socket interface to the SGSE, using application identifiers;
- Send/receive data on the Virtual Socket interface;

EXT3: Connectivity Agent ←→ TCP/IP

This interface operates on the data plane. It is a standard socket interface to the TCP/IP protocol stack as provided by all the existing operating systems. It provides the following operations:

- Open, bind and close a socket to the TCP/IP protocol stack using either UDP or TCP transport;
- Send/receive data on the real socket interface;

EXT4: TCP/IP ←→ Link Interface

This interface operates on the data plane. It offers a standard and common LLC or IEEE 802.2 interface to the Network layer and provides the following operations:

- Send/receive data on Network-to-Link layers interface

EXT5: Link Interface ←→ Link Wireless Access

This interface operates on both the data and control planes. It actually depends on the set of commands for accessing each technology and the terminal implementation. For example, to access an LTE mobile terminal, AT commands as defined in [95] are used. Section 6.5 provides the mapping performed in the Link Interface between the MIS primitives and this interface for an LTE/EPS access.

4.2.3.2 Internal Interactions

These interfaces enable the cooperation between the different entities of the framework.

- INT1: User Interaction ←→ Connectivity Agent

This interface is used by the CA to trigger the user for some information like his preferences, credentials or decision confirmation.

- INT2: Cognitive Manager ←→Generic Service Enablers

  This is an interface internal to the CA, used by the CM to orchestrate the operation of the GSEs, or by the GSEs to report execution results and monitoring information.

- INT3: Generic Service Enablers ←→MIS Function: MIS_SAP
- INT4: MIS Function ←→Link Adaptation : MIS_LINK_SAP

  INT3 and INT4 form the abstract interface between the GSEs and the Link Interfaces, relayed by the MISF.

- INT5: Connectivity Agent ←→Cross-Layer Agent
- INT6: MIS Function ←→Cross-Layer Agent

  INT5 and INT6 form the interface to the CLA, used by all the CCF components to store and retrieve knowledge from the LIB.

These internal interfaces are detailed (primitives, function description and parameters) in ANNEX A.

4.3 Subsystems description

Building on the previous sections and the definition of the internal and external interfaces between the various entities identified in the CCF, illustrated in Figure 41, this part describes the framework internals and provides more details about the behaviour of its components, specifying their operations and interactions. The cross-layer and shared knowledge functionality is handled by the CLA and its LIB. The Link Interfaces and the MISF take care of the device and technology specificities and hide them behind an abstract interface. Three GSEs, the first one dedicated to network access selection, the second one to the terminal connectivity and the last one to the application sessions management hold the necessary operation of the MT. Finally, the CM coordinates their actions, to obtain an autonomous behaviour, lightly assisted by the human user through the UIA. For each component, the corresponding section defines which optimizations it provides, what other components it interacts with and what it brings to the global system.

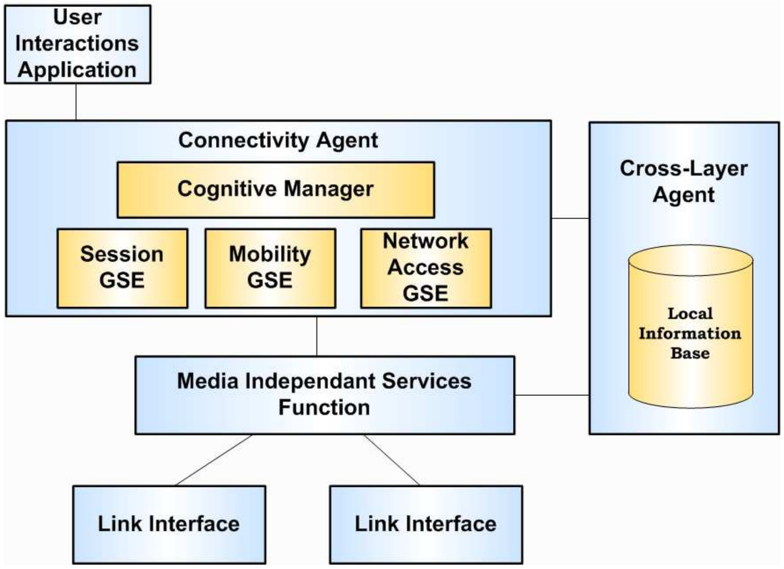


Figure 41: Internal components of the CCF

4.3.1 Cross-Layer Agent and Local Information Base

The Cross-Layer Agent (CLA) is responsible for gathering the data from the devices and Link Layer technologies and from the upper layers, for aggregating them in the Local Information Base (LIB) and for providing them on request when needed to the other CCF components.

Several cross-layer approaches exist in the literature. The objective is to enable communication between different layers, which may or may not be adjacent in the OSI model, and allow them to cooperate in an efficient way in order to optimize the global operation of the system. A first approach consists in establishing direct interfaces between all the different layers present in the system. This approach is not very efficient as it results in reduced system performance and sub-optimal overall optimization. Another approach consists in introducing a transversal cross-layer engine that communicates with all the different layers. The cross-layer engine being externalized, the complexity is kept outside of the different layers, which is better for performance concerns. A typical application is the Management layer in the ITS Station model presented in Section 2.6. This approach also offers a great advantage for maintainability. However, it strongly depends on a single component responsible for all the cross-layer interactions and may easily become a bottleneck to the overall performance if it has to assume the whole cross-layer operation.

The approach adopted here is a hybrid of both solutions to combine their benefits. A cross-layer engine, the CLA, is introduced to work as a local Information Server. It is responsible to manage a local storage, the LIB, making it accessible to the other components in the framework, namely the MISF and the sub-components of the Connectivity Agent, while preserving its integrity. In that objective, each CCF component provides the relevant information to the CLA after performing a new action. The CLA stores the configuration information and learns dynamically all the parameters and policies along the MT operations. It is the component responsible of the learning process in the CCF. Based on the feedback from the user and the physical environment received through the other components, it analyses the values of parameters and policies that have been applied to determine the rewards and sequences of {actions, state transitions} that can optimize the operation of the CCF. This part has been studied here at the concept level and left for further study in the future. The CLA is able to provide this information back to the other CCF components on request in a generic way. In parallel, direct interactions between the adjacent components (CM, GSEs, MISF, Link Interfaces) are kept to transfer events and commands, according to the MIH model. It distributes the complexity and ensures a quick response of the overall framework to changes in the external environment.

Local Information Base

The LIB is the shared knowledge source for the whole system. It contains all the data relevant for an optimized operation of the framework, as shown in Table 7. 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 (MIIS/ANDSF) 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. Table 7 gives a list of the main parameters stored in the LIB. It is organized into layers, according the CCF architecture, from bottom to top. The first column indicates the data group, the second column the type of parameters it holds and the third column gives the list of the main parameters contained in this group. Except for the MT internal information, the LIB is able to accommodate several instances of each group of parameters.

Data Group	Type	Description
MT internal information	Status	Battery consumption, speed, geographical location, time, device capabilities (multimedia, multicast or MBMS support), power management features.
Link information	Status	LinkId, Link Address (MAC address or other), technology, link status, signal quality (SNR), available bandwidth, throughput, Packet Error Rate. A complete list of the parameters defined at WLAN and LTE interfaces can be found in [78].
Connection information	Status	Net Id, Network name (APN or SSID), Link type (technology) and Id, user approved/rejected, connection status, EnvA, address expiration time, routeability, capabilities (multicast, language, etc.), attachment policies: type of mobility supported, security attachment type (L2/Network), credentials data
NAGSE decision history and parameters	Configuration + Learning	Decision policies: link reward for each parameter per application and access network Decision inputs : list of normalized parameters Decision output: ordered list of (access network, score) pairs.
Session parameters	Status	Session Id, Application Id, connected Net Id, PA,

		EnvA, QoS obtained, Port numbers, destination address, transport protocol, DTN mechanisms required, DTN buffering status
Application QoS and network requirements	Configuration	QoS class, bit rate, priority, reliability, latency, maximumTransmissionUnit. Preferred technology, Multicast (MBMS).
User preferences	Configuration	Technology precedence (order, conditions) Net Id, approved/rejected, L2 authentication credentials.

Table 7: Parameters in the LIB

Even though it contains a large number of parameters, the LIB is expected to remain at a reasonable size well under the memory present in the mobile devices currently at the disposal of mobile users. It is more efficient if set as persistent memory.

In summary, the CLA, with its LIB, brings to the framework the capability to learn, store and distribute a common knowledge about the terminal internals and its environment. This knowledge is shared across the different layers of the framework.

4.3.2 MIS Function and Managed Interfaces

This section describes the Media Independent Services Function (MISF) component, enhancing the IEEE 802.21 model in the terminal nodes and contributing to the framework shared knowledge with its direct access to the cross-layer LIB. In its second part, it introduces the operation performed by the Link Interface components, whether they manage a wireless access or another device such as a power source or a positioning system located in the mobile.

Abstraction Layer

The MISF is an abstraction layer responsible for dealing with the wireless multimodality of the terminal. It is a key component of the connectivity optimization process, as it provides the means for the abstracted interaction between the radio access and the upper layers in order to accomplish cross layer functionalities and underlying technologies transparency. It is based on the IEEE 802.21 MIH services, but is not restricted to handover; it fully manages the wireless accesses and other devices in the terminal. It has been named MIS because it does not strictly comply with the 802.21 standard and has been designed with a different objective and methodology. It only reuses the Media Independent or Abstraction Layer concept to provide services to upper layers, hiding the specificities of the lower layers. It extends the media independent signalling functionalities by supporting a large set of functions, i.e. power up, power down, wireless access scanning, reporting of adaptive and dynamic parameters but also available access network monitoring, system statistics and status retrieving, resource configuration with a certain level of QoS, setting and getting identities, setting security features, handling power sources, positioning, multimedia support or enabling multicast and broadcast services, etc. Any external device available in the mobile can be managed by this component.

Moreover, it provides an abstract interface to the CLA component which handles the LIB. It provides to the local storage the network or device information received, contributing to the system learning. It retrieves L2 credentials or link specificities when requested to issue a command to the lower layers, avoiding that the upper layers get involved with the device details and that the CLA offers too many specific interfaces towards the various devices.

This enhanced MISF operates primarily in the control plane. Its objective is realized by using abstract interfaces with both upper and lower layers, namely the GSEs on one hand and the Link Interface components on the other hand. For the wireless access management, the MISF interacts with the GSEs to handle the parameters for access network selection, connectivity control or session setup, reporting measurements or providing information about the availability of the wireless accesses and the dynamic variations of the radio channel. It forwards downwards the received commands which carry the setup of the wireless access and upwards the dynamic information provided by the device. In order to save on the processing power and battery consumption of the terminal, the operation at and above the MISF is event-triggered only. Except inside some of the lower layers which need it to assert those events, the framework does not rely on the periodic scanning of any parameter. The MISF is completed by a data plane sub-component which enables the data transfer between the wireless access protocols at Layer 2 and the Network Layer, exposing an upper interface at IP level identical to the LLC (IEEE 802.2) protocol [79]. The LLC corresponds to the upper part of the OSI Data Link layer, identical and used by all wireless 802 protocols, while the MAC is the lower part dependent of the medium used. The LLC provides several operational modes for the transmission of data packets: unacknowledged connectionless, connection-oriented and acknowledged connectionless. For IEEE 802.x wireless technologies, this part is transparent, since they already use the LLC, but this component becomes very interesting for cellular or other technologies (Digital Broadcast for instance) which often run their own specific protocols and offer different L2 interfaces. The MISF data plane allows them to be seen as a standard WLAN access by the Network layer.

The MISF also enhances the 802.21 standard with a capability to manage devices which are not network interfaces. Parameters such as geographical location, current time, battery level or any other sensor data can be reported to the GSEs which, in the reverse direction, can configure the functioning of these devices. For that objective, the MISF uses the same set of primitives but with different parameters.

Common operation of the Link Interfaces

These components make the link between the MISF and the technologies device drivers. There is one Link Interface per type of device, completely specific to the device. Its main function is to translate the MIS commands and forward them to their corresponding device. It acts as the endpoint for parameters retrieval from the device. It receives the configuration MIS primitives and executes internal procedures to be able to report measurements or subscribed events. In these procedures, it analyses the parameters accessible in the device to provide the requested measurements or send event notifications. In that objective, it may have to schedule some periodic monitoring of 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 handovers.

Link Interfaces for the wireless accesses

The wireless accesses considered in this study are WLAN as a contention-based technology and LTE as a coordination-based technology, as best representative of the technologies used for communications in modern smartphones; ITS G5, Bluetooth, Infrared or any other technology could be managed in the same manner. Technologies by themselves remain unchanged, except when new commands may be needed to comply with the requirements of one of the services, which has not been the case in this study.

The Link Interface components control the selected technologies, handling connection-oriented and connectionless constraints. These components support the operation of an abstract interface, enabling the operation of the framework with any upcoming access technology. One component is included for each technology supported, which convert the generic primitives and configuration parameters received from the upper layers into commands understandable by the underlying heterogeneous technologies, and vice versa. This conversion has been addressed for IEEE 802.X, 3GPP UMTS (Universal Mobile Terrestrial Service) and 3GPP2 media by the MIH standard, but not for the 3GPP LTE. Section 6.5 describes the mapping between the MIH primitives and the EPS/LTE procedures that has been submitted to the standard. The Link Interface components include these mappings, plus those necessary to comply with the extended functionality of the MISF.

Link Interfaces for the other devices

A mobile terminal, whether it is a laptop, tablet, smartphone or car system, includes devices other than the wireless interfaces. A non-exhaustive list would include positioning systems (e.g., GNSS), power supplies (solar cells, batteries, power plug or car engine) or sensors such as RFID tags, light sensors... In the same manner as the wireless interfaces, these devices can be controlled and monitored using specific drivers in the terminal. When related to the mobile connectivity, they are thus candidate to a coordinated and integrated control through the CCF and the MISF, provided the availability of a Link Interface component to translate the generic MIS primitives into their own set of commands.

Using this feature may prove very interesting in future terminals equipped with sustainable energy for example. Imagine a mobile equipped with two different energy sources: a battery and solar cells. The solar cells are sufficient for accessing the WLAN, but not for the cellular, which mandates the use of the battery. Executing a handover from WLAN to cellular means that, simultaneously, the power source has to be changed as well. With the CCF and the extended MISF control, it is possible to command both switches (network and power source) from a single intelligence in the system.

MIS primitives

The 802.21 set of primitives is simplified to make this component generic. The only primitives defined are Link_Action, Link_Configure, Link_Report and Link_Information. They are used with various sub-types to clarify the expected objective and carry parameters as required. This brings an additional level of flexibility to the system, since supporting a new function consists only in defining a new sub-type with its associated parameters. The three services that were described in Section 2.2 are used here: MICS for requesting actions and configuring, MIES for receiving reports and events, MIIS for storing and retrieving the local information.

The paragraph below gives a list of these primitives with their sets of sub-types corresponding to this study. The details of the parameters associated with each primitive and sub-type can be found in Annex A.3**.**

- Link_Action.request/confirm:
  - Monitor network interfaces (All / per technology) – see Note 1
  - Attach / Detach to selected network / PoA
  - Execute L2 authentication
  - Retrieve user context parameters from the devices (network address, location, time ...)
  - Activate data resources
- Link_Configure.request/confirm:
  - Control link measurements (periodic report request, event subscription)
  - Control session measurement
- Link_Report.indication:
  - Network event
    - New access detected
    - Network rapidly decreasing
    - Network lost
    - Network congested
    - Network performance unstable
  - Measurement event
    - Periodic measurement report
  - Battery event
    - Low level battery detected
    - High level battery detected
  - Location / Time event
    - New location detected (through sensors: office, home, car, store. ...)
    - Scheduled time event (used to get notified of a subscription rate change)
- Link_Information.request/confirm:
  - Get parameters from the LIB
  - Set parameters in the LIB.

Note 1: This function includes as well the request for reporting parameters such as the network capabilities (language, multicast ...).

In summary, the MISF and the Link Interfaces bring to the framework the capability to manage, in an abstracted and flexible way, the various network interfaces and devices present in a terminal in order to achieve an optimized connectivity.

4.3.3 Generic Service Enablers

The Generic Service Enablers are the key elements of this framework. As their names indicate, they provide services in a generic and specialized manner to the upper layers above the CCF, the applications on the data plane and the NS and UIA on the control plane. In the Connectivity Agent, the GSEs are complemented by the Cognitive Manager, which orchestrates their actions and brings an additional level of autonomy to the whole system. As a result from previous sections, at least three generic service enablers are necessary to enable an efficient connectivity control across heterogeneous and independent networks: network access selection (NAGSE), connectivity and mobility (MGSE) and session (SGSE). Other services like GeoNetworking control, multimedia optimization, security, adaptability of the system for energy efficiency, positioning could easily be added to this list.

The GSEs 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 Network Services, and provide a unique means to translate them to the lower layers. They can query the LIB for aggregated relevant cross-layer metrics and provide them to the upper layer services. They act as MIS users and hide the MISF interface to the legacy applications and Network Services. They use the same concept as the ITS Facilities Layer (see Section 2.6) or the services in the OMA Connection Manager (see Section 2.4).

Table 8 summarizes the distribution of the networking procedures between the three GSEs defined in the CCF. These procedures correspond to the main steps of the functional view of a mobile terminal operation, as described in Section 4.1.1. Each GSE is specialized in a specific task but handles several related functions. This avoids the multiplication of functional blocks and a useless complexity of the framework. Each function listed in the table is explained in one of the next sub-sections which describe the three GSEs.

GSE	**Connectivity function**
NAGSE	Monitor availability of access networks
	Network selection algorithm
MGSE	Network interfaces and link management
	Network access control : registration / L2 authentication
	Network address retrieval (EnvA)
	Receive and filter network events
	Keep track of current location and connectivity (mobility management)
SGSE	Network authentication
	Session management: PA and real sockets on TCP/IP
	Reservation of resources with the required level of QoS
	Packet transfer; map virtual socket to real socket
	Execute light DTN procedures (packets buffering) during handovers

Table 8: Distribution of networking functions between the GSEs

4.3.3.1 Network Access GSE

The Network Access Generic Service Enabler (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.

For the discovery and monitoring of available networks, it uses mainly the information received by the network interfaces, either to identify the availability of a known network or to learn the necessary system information from an unknown network: RAT, network name (Service Set Identifier or SSID for WLAN, Access Point Name or APN for cellular), signal quality and when possible, the capabilities and available bandwidth. Information about network capabilities (including characteristics beyond the individual point of attachment) can be very useful for the selection of an access network and the framework provides this service. This information is stored in the LIB and later used for the network access selection. In some networks, an information service such as a MIIS server or an ANDSF may be available (see Chapter 2). Whenever possible, the NAGSE is configured with their addresses, or with the means to be able to discover them, and requests more information from these mechanisms in the network to assist the CCF operation in a more efficient manner. This allows learning and storing the necessary information about the available but yet unknown access networks.

The access network selection algorithm in the NAGSE may be invoked from several different states of the system:

- when the terminal starts and needs to identify the initial network to attach to, without any running application;
- when a new application starts, it checks whether the connected network is suitable. If several are available at once (in case of multi-homing for example), it evaluates which one is the most convenient;
- during mobility phases, it is triggered to provide a choice of target networks to which the handover can be executed.

The objective is to apply an algorithm to a set of parameters and derive a network configuration, in the form of the preferred ordered list of access networks, according to known policies and the suitability to transfer the user data traffic. The criteria introduced by the policies govern the following metrics: better coverage, connectivity stability (network availability, routeability, foreseen number of handovers), load balancing, energy efficiency, user preferences, application requirements in terms of bandwidth, QoS, technology or network support (voice call, MBMS...), capacity stability and network security.

According to the survey performed in Section 2.4, algorithms based on MDP or Q-learning approaches give the best results. However, they imply very long converging delays, and thus are not directly suitable for a dynamic operation. More classical algorithms, based on MADM methods, evaluate a finite number of alternatives and thus provide results in a more reasonable time period. In particular, the most widely used, the SAW algorithm is simple, converges in a limited amount of time and requires a reduced processing time since only one score value per access network has to be computed. Moreover, it is flexible and can accommodate a larger set of parameters, resulting in a more adequate decision. Several papers in literature, [80] [59], have proved that, even if not optimal, it provides quite acceptable and stable results.

Before executing the algorithm, the NAGSE retrieves the values of the parameters and their link rewards from the LIB. The input parameters considered are split in three main groups: *(i)* user preferences, *(ii)* application and QoS requirements and *(iii)* network/ environment status and properties, and are listed below.

- user preferences (network cost and user-defined precedence list of networks, depending on the timing and rates of communications),
- suitability of the network for the application,
- status of the access (SINR),
- QoS of the access (available bandwidth, jitter, BER, stability: percentage of variation over time),
- mean battery consumption compared to current battery level

- size of the cell geographical area compared to the MT (or user) speed,
- network route metric,
- knowledge of credentials.

The application requirements on QoS (bit rate, reliability, latency, priority) and on other parameters are translated into link reward values associated to each parameter. They are determined in an offline process which is not detailed here and stored in the CLA (see Section 4.3.1). To simplify this process, applications are grouped according to their QoS requirements, with the same reward values in each group.

In order to improve the performance of the whole decision procedure, the NAGSE first performs an initial check to compare the value of the application suitability with the candidate access networks to be evaluated. If the link reward is equal to 0 (which means that the access network does not match the application), the network is eliminated from the list of candidates. The same check is performed on a few other parameters such as the available bandwidth vs. the minimum acceptable value, the network coverage vs. the terminal speed or the user's preference in terms of network. For instance, if the user has indicated that the application should not be used with that access network.

Next, the SAW algorithm is executed. MADM methods can be split into two processes, one for the normalization of the network attributes involved in the decision, the second for the aggregation of these attributes. In the first step, each parameter is transformed in a numerical value which is normalized afterwards, to be comparable to the other parameters. This is straightforward for measurements such as SINR or current battery level. More complex or subjective parameters are transformed using utility functions. When the normalization process is linear, it transforms the measurement according to the formula [81]:

$r_{ij} = \frac{x_{ij}}{x_j \max}$ for a benefit and $r_{ij} = 1 - \frac{x_{ij}}{x_j \max}$ for a cost, where

r_{ij} is the normalized value of the j-th parameter of the i-th network

x_{ij} is the measured value of that parameter

$x_j\max$ is the maximum possible value of that parameter

The output of the normalization process is a *NxM* decision matrix. Considering a set of 12 evaluated parameters and 5 candidate access networks per decision, this matrix contains 60 elements. It is thus possible to store the history of former decisions in the CLA and look it up at this point of the process, to check whether a decision under the same conditions has already been made. Many well-known search algorithms exist which can perform this task. If a valid entry is found, its result is adopted and the decision process stops.

If no valid entry is found in the CLA, then, a decision has to be computed. The score S_i of the current context for the i-th target network is determined thanks to a single calculation as

$$S_i = \sum_{j=1}^{n} w_j . r_{ij}$$

where w_j is the link reward of parameter *j* for the application and the access network considered. The terminal start-up case, where no application is yet running, is considered as a *NULL* application with its specific rewards values. In order to avoid the Ping-Pong effect and privilege the existing connected network, its link reward values obtained from the CLA are slightly increased. The score obtained as a result of the algorithm by each access network allows to determine an ordered list of (access network, score) pairs. This result is forwarded to the CM for the final decision making. When the score is not high enough, it means that the confidence in this score is not sufficient enough and a validation from the user, prompted by the UIA, is necessary to confirm the decision.

Because the algorithm depends on a combination of a discrete number of parameters and link rewards in a limited number of cases, it converges very rapidly. The major contributor to the workload of this process is indeed the offline definition of the link rewards associated to each parameter in the CLA. Moreover, this algorithm is very flexible. It is easier to add a parameter than with “if then else” policies because it only requires that the new attribute get allocated link reward values.

In summary, the NAGSE brings to the framework the capability to characterize the network environment in a very precise manner and to select the best access network for each application, taking into account several independent criteria in a flexible algorithm.

4.3.3.2 Mobility GSE

The Mobility Generic Service Enabler (MGSE) is the most commonly developed part of the Connectivity Agent. Its role is to take care of connectivity related services, including network interfaces and link management, network security at L2 level, network address retrieval (EnvA), reception and filtering of network and device events, keeping track of current location and connections. The process of the MGSE is completely independent from the mobility protocols (MIP, PMIP, SIP or others) that may be available in the network and/or the terminal and would run as part of the NS described in Section 4.2.1.

The MGSE executes the commands received from the CM to power up or down, activate or deactivate and wake up the network interfaces and link connections, including the handling of registration and L2 authentication procedures when required. The result should be the establishment of a signalling path to perform network authentication, registration and, if relevant, mobility binding.

When a network has been connected at Layer 2 level and the system switched to the new interface, it is responsible for obtaining the EnvA described in Section 4.1.3. The MGSE operates at the same level as the MIF API that is being proposed at the IETF, but benefits from the abstraction of the MISF. Its functionalities include the update of the network interfaces configuration and of the routing table’s content, to enable the later correct operation of the SGSE. It provides the information that it has collected during the activation or connection phase, including its final result, to the LIB, allowing the framework to keep track of the current connectivity and system status.

Another major function consists in subscribing to, receiving and balancing the various events received from the whole system, in order to decide whether they are relevant enough to trigger a handover and improve the connectivity. When the answer is positive, the event is forwarded to the CM which takes the appropriate actions. The major part of these events is originated from the Link Interfaces through the MISF and includes:

- network event (detected, going down, lost, congested, data rate or connectivity unstable),

- location (through sensors or GNSS).
- battery level crossed a threshold (upwards or downwards),
- time (change of subscription rates),

Other actors in the mobile and external environment may execute an action that triggers a handover to another access network:

- applications or service (start/stop),
- Networking Services component (split current traffic between two networks to maximise the bandwidth, expiration of the network address, balance network load, battery power save mode initiated and so forth),
- the mobile user (disable a technology).

Most of the latter events, out of scope of the MISF and the MGSE, are notified directly to the CM by the appropriate component.

In summary, the MGSE brings to the framework 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.

4.3.3.3 Session GSE

The Session Generic Service Enabler (SGSE) deals with aspects related to the management of the data sessions opened by the applications, the network authentication or potentially Media Content Adaptation. It also takes care of the availability of resources corresponding to the applications QoS requirements. In the data plane, this component performs the address translation between the PA seen by the application and the environment IP address (EnvA) seen by the network. For the duration of handovers, it executes a buffering mechanism inspired from DTN techniques, storing the packets received from the application until a new connection is safely established. It is the component responsible to re-start the TCP connectivity when it has been broken due to the change of connected network.

Control plane operation

When network authentication is required in a newly accessed network, the SGSE is asked to start a generic HTTP session to be able to enter the user credentials that it has retrieved from the LIB and allow further communications on that network. When the credentials are not known in the LIB, the UIA is triggered to prompt the mobile user.

In the CCF, the applications are identified by their protocols and the well-known port numbers they use. When an application starts, it opens a socket on the Virtual Socket interface, providing the address and port number of the destination. From the application, this interface is seen as the standard socket API. However, it does not output to the transport protocol, but rather to the Connectivity Agent. After retrieving the application requirements from the LIB, and checking with the NAGSE that a suitable connection is ready for this application, the CM asks the SGSE to set up a new session. The SGSE first allocates a PA to the application. Here, to simplify the process, the PA is chosen equal to the initial EnvA used when the application starts. Then, the SGSE opens a real socket on the TCP/IP socket API, with the same properties as the virtual one, providing the EnvA as source address. It binds the PA to the EnvA and, when possible, it requests the allocation of resources corresponding to the required QoS through the MISF.

During an inter-domain handover, it copes with the break of the session by automatically re-establishing a fresh one through the new access network. It prepares and establishes it in the same way, updates the binding between the PA and the new EnvA 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. Establishing a fresh TCP session brings an additional short but not significant connection delay (initial three-way handshake) to the total handover delay of the framework. Its order of magnitude is in dozens of milliseconds, depending on the wireless access network.

Data plane operation

During normal operation, the SGSE receives the packets from the application, translates locally the application PA to its bounded EnvA, does the same for the port numbers if necessary, and forwards the packets towards the real socket.

When an inter-domain handover has been decided by the CM, the SGSE is called to start a buffering the upcoming packets, in the same manner as done in the DTN mechanisms. The packets received from the application are not forwarded to the transport protocol, but rather to a First In, First-out (FIFO) queuing system which stores them for the duration of the handover and establishment of the new TCP connection. Considering that a handover is a fast operation, the amount of temporary storage required is not very important, as shown below in a very pessimistic case. Considering

Data rate at the application: 1 Mbit/s

Duration of the handover: 6 seconds (see measurements in Section 6.1)

Number of simultaneous applications running: 5

We can evaluate an order of magnitude

Amount of storage needed per application: 6*1M = 6 Mbits or 0.75 Mo.

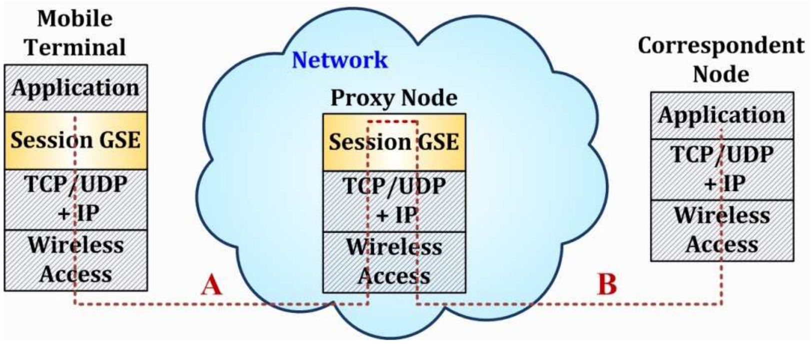
Amount of storage needed for five concurrent applications: 30 Mbits or 3.75 Mo.

Provisioning a temporary storage of 5 Mo seems reasonable, as it corresponds to less than 1% of the memory available on current smartphones (512 Mo of RAM memory generally) and will for sure be over passed in the next years, if not months.

The next operation is to transfer safely the data flow to the new connection and socket. Thanks to the PA and the packet storing mechanism, this operation is completely transparent to the local endpoint of the application which sees the same address and same port number. However, the session continuity has to be ensured at the Correspondent Node as well, whether it is a server or another mobile terminal, and despite the break of the TCP connection. Two mechanisms have been analysed as pictured in Figure 42.

Option 1:

Intermediate proxy



Option 2:

End-to-end

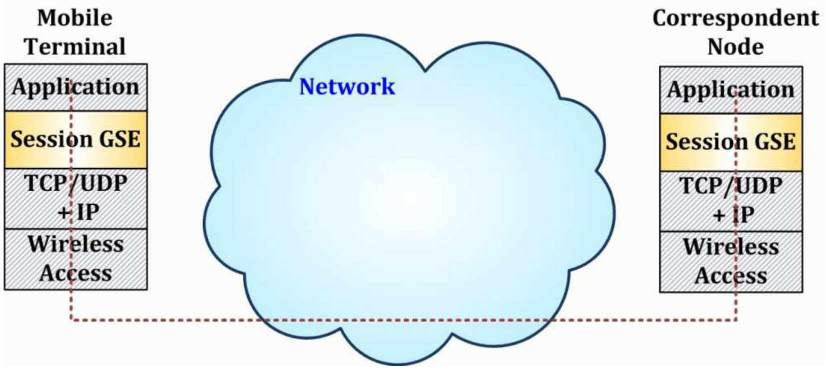


Figure 42: Communication with the correspondent node

In option 1, a scheme similar to those found in the literature (see Section 2.3) has been adopted. A proxy node is introduced in the network that executes the SGSE functionality and hides the MT actual connection to the CN. In this case, the CN is unchanged. The proxy node can maintain the connection B during the time the connection A is broken (buffering the packets received from the CN in the proxy SGSE component). The session B is completely unaffected by the change of session A, but both sessions have to be re-synchronized (e.g. for TCP congestion window) to avoid over-flowing the proxy node after the new session has been established on the connection A. The drawback of this solution is that it is less scalable and it introduces some delay in the data traffic, especially for long RTTs (Round Trip Times) on connection A or B. Moreover, it requires an organization, such as a service provider, to setup and maintain the proxy node, with associated costs an constraints. Another issue is to ensure that both endpoints (MT and CN) can safely trust the proxy node.

In option 2, 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 more scalable, as there is no intermediate node, but requires a change in the remote node, similar to an Operating System update.

In both options, the network infrastructure itself is not modified and the framework remains identical in the mobile node. A single message signalling between the two SGSEs (secured with an exchange of authentication keys) notifies the CN when the initial session is established and when the IP address has been changed, to trigger the update of the session addresses mapping. The operation is performed at a sub-layer 5 (Session layer) level, so it does not introduce any security impact on the network level connection. All the existing mechanisms are maintained. Because a fresh TCP session is established over the new network connection, this is also compatible with firewalls and other security devices. The overhead introduced by this mechanism to the data traffic is limited to the address translation inside the terminal. On the control plane, the overhead consists in a short secured signalling over the new connection to update the binding on both sides and the establishment of the new TCP connection, which both contribute to the handover delay. For its better scalability and security support, it is the option 2 which has been chosen here.

When the new session is ready, the SGSE stops buffering the packets and empties the queue by sending its contents to the new socket, before restoring the application data transfer.

In summary, 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. It takes care of all the sessions, opens and repairs as necessary the socket connections, while enabling the continuity of the connectivity with the CN.

4.3.4 Cognitive Manager, leading the Connectivity Agent

The architecture proposed in Section 4.1, with the components described so far, provides all the functions necessary for the user to operate manually his mobile terminal under normal conditions and with sub-optimal efficiency. As demonstrated in Section 2.1, when the application that he is executing freezes or stops because an inter-domain handover has occurred, the user currently has to re-start it manually. Furthermore, the blueprint of Autonomic Systems (AS) has been presented in Section 2.5. Both architectures show some analogies, hence it appears interesting to map the CCF in the MT to the autonomic blueprint of a managed network and provide autonomy to the mobile system, removing the hassle of manual actions from the mobile human user.

Figure 43 shows how the AS blueprint has been mapped to our framework to provide a layered autonomous system. Its analysis from bottom to top shows that the wireless access technologies and the other devices are the managed resources inside the terminal. The Link Interface components and the MISF serve as the manageability interface, monitoring and controlling the resource behaviours with the MIS Event and Command Services. In the autonomous control loop, they implement the sensing function by providing the access network status or other parameters in a background and lowest priority task. The monitoring function collects information and aggregates or compares them to thresholds until an abnormal condition is encountered. The actuators in the Link Interfaces are indeed requesting actions such as power up or attach commands. Directly interacting with the MISF are the GSEs, each of them dedicated to a specific role and analogous to the IAMs. They need to be coordinated to provide an integrated autonomic behaviour. This integration is performed in the Connectivity Agent, which includes both the CM and the GSEs, either through one of the autonomic properties, i.e. self-configuration, self-healing and self-optimization, or even across these properties. In our architecture, the CM plays the role of the autonomic system controller, orchestrating the self-management functions of the MT to increase the level of efficiency of the global framework. Its internal functionality is described in the next paragraphs. It is assisted by the UIA which provides a simple user interface to the human owner of the terminal, playing the role of the Manual Manager in the AS ontology. The CM obtains information and triggers actions from the GSEs. It coordinates the actions of the

GSEs according to its own state and the events received. Any decision or execution is generated by the CM. The knowledge source is mapped to the CLA and its LIB.

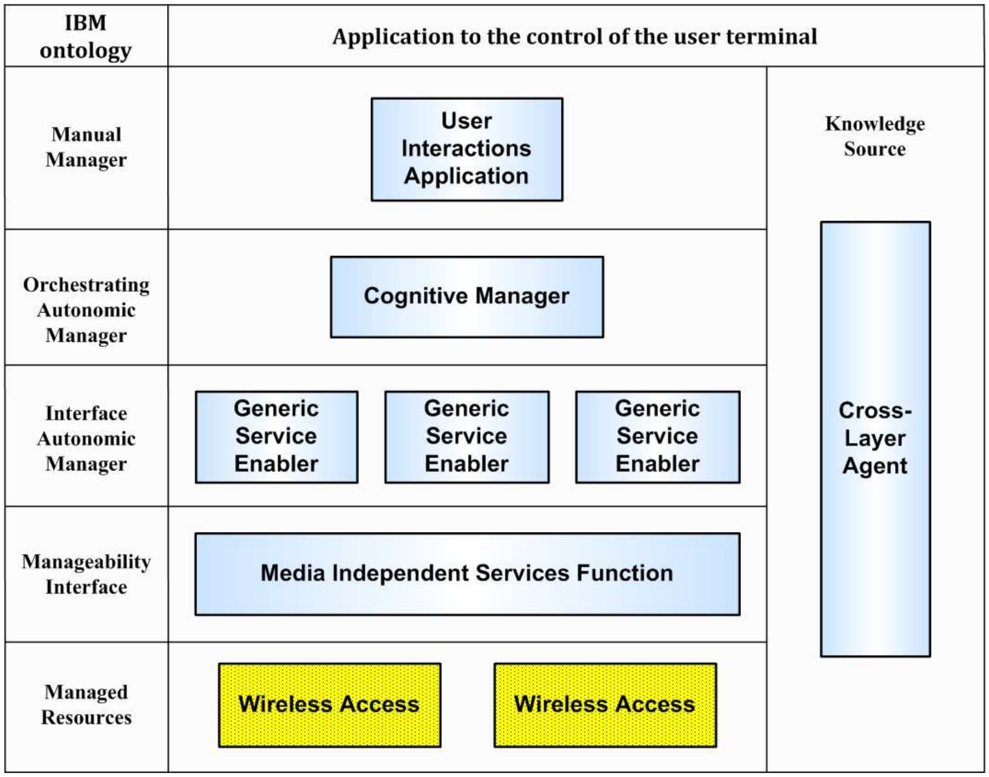


Figure 43: CCF layered model

From an internal point of view, the CM implements the system control loop that was presented in Section 2.5 and includes the following functions:

- monitor the system behaviour by subscribing to MIS events and offering external interfaces to the applications, the Network Services or the mobile user;
- analyse the received events informing about environment changes, by using the parameters provided with the event, external knowledge from the CLA or result of further investigations requested to one of the GSEs. The user may also be solicited though the UIA;
- plan the best action to take, and make decisions according to its current state;
- trigger the execution of these decisions by sending commands to the GSEs, that will be forwarded through the abstract interface to the various devices. Here, the actuators (GSEs, MISF) are responsible to update the global system knowledge with the results of their actions.

In summary, 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.

4.3.5 User Interactions Application

The User Interactions Application (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 is used to obtain the user preferences at configuration time or making and validating 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.

The learning process is twofold. Firstly, at configuration time, the UIA is used to retrieve information and preferences from the user and build a User Profile. The preferences arrange the technologies based on pre-defined conditions: subscription rate timings, plans and policies, security level of a specific location (at home, at office or in well-known stores) or network load. They may also prevent some applications from automatically using specific interfaces or networks (e.g., a web radio with 3G, a proprietary VoIP on a mobile operator network or a bank application on a public network). The description of the known and approved networks provides the CCF with the knowledge of their identities and Layer 2 credentials (WEP or WPA keys for example in WLAN or virtual SIM card for cellular networks). Default values for every field are pre-selected in order to have selections where the user did not make any choice. As mentioned before, this information is the basis of the shared knowledge of the CCF, saved persistently in the LIB. Secondly, at run time, the UIA can be queried when the autonomous behaviour does not have a sufficient confidence level in its connectivity decision or when security credentials are needed to access an unknown network. The missing information is prompted to the user, and then stored in the LIB as part of the dynamic rules, according to the learning process of the user decisions. In the reverse direction, the user can check the configuration information in the LIB through the UIA and alter the value of the parameters when he prefers to change them.

The configuration information entered by the user can be completed by policies received from the network. This applies to common application preferences in terms of QoS or networking, which could be retrieved from a dedicated server in the cloud, because entering them one by one would be tiresome for the user. Some operated mobile networks may also mandate that the connected terminals retrieve their own policies to provision network identifiers and rules in addition to the user settings.

In summary, the UIA brings to the framework the capability to receive directions and knowledge from the mobile human user.

4.4 Summary

In this chapter, the proposed solution has been presented from a global then detailed point of view, showing the operation, the internal functions and external interactions of the framework. According to the requirements defined in Section 3.2, the resulting layered system modifies only the MT, leaving the network unchanged. It revolves around three main principles that guarantee a simple and flexible architecture, which could be summed up in a modification of the terminal operating system. The first principle is to hide the heterogeneity and diversity of the devices and access networks behind an abstract interface which facilitates a range of services wider than handover management. This is achieved by the MISF and the Link Interfaces. The second is to provide coordinated generic service enablers that can take care of dedicated operations. They enhance 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 NAGSE, the MGSE, the SGSE and the CM. The third is to share the knowledge about the terminal context and its environment between the different components in a cross-layer fashion. This is achieved by the CLA which stores the configuration, policies and status of the framework in the LIB. To enhance these principles and their efficiency, an additional component, the CM, has been introduced, which allows the terminal to roam autonomously across the different access networks, and relies on human interaction only when the level of confidence of its self-management algorithms is too low. Each of these components has been detailed from a functional point of view, highlighting its contribution to the whole system. As required in Section 3.2, the defined components are based on existing concepts and technologies, enhancing them when needed. They combine their individual actions in order to bring the whole framework to its expected level of resilience and efficiency. The benefits and impacts brought by this framework on the terminal must be evaluated in the different scenarios proposed in Section 3.1 and from a global point of view. This is the objective of the next chapter.

CHAPTER 5 - MODEL VALIDATION AND RESULTS

The previous chapter has presented an architecture to make the relevant static and dynamic characteristics of the mobile terminal environment available to the upper layers in a unified way. Consequently, the provided CCF framework is able to combine and fine-tune existing networking techniques in the terminal to overcome connectivity issues when moving between independent non-operated networks. This chapter attempts to evaluate the efficiency of this architecture by an analysis applied to the scenarios defined in Chapter 3 and validate the chosen technological scheme. A simulation model has been developed and is presented in the first section. Its objective is to demonstrate the capabilities and the benefits of the cross-layer approach. The second section describes the scenarios and conditions of the simulation executions. Their results are discussed to evaluate the global operation of the framework. Possible extensions of the model to enhance its operation are highlighted.

5.1 Simulation of the CCF

In order to assert the benefits of the framework described in Chapter 4, its behaviour has been tested by simulation. In that aim, a simulation model has been developed and implemented. The first part of the section describes the objective of the simulation, then the chosen environment and its selection criteria. It is followed by a presentation of the reference model prepared, which allowed to collect the data presented in Section 3.3. Finally, the CCF model used for the validation and its implementation choices are covered.

5.1.1 Evaluation Objectives

The objective of the simulation is to assess the validity of the CCF and evaluate its benefits. It focuses on adding the CCF components to a wireless multimode terminal and moving that terminal in a sample network which includes two heterogeneous access networks (WLAN and LTE) and an Internet connection. By doing so, it evaluates the impact of the framework in the context of the scenes described in Section 4.1.3, using a selection of common applications (FTP, Web) completed by “Ping” trials. As mentioned in Section 3.2, 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. Due to the wide scope of the CCF, the objective of the simulation test had to be restricted. Since the abstraction framework was experimented in real systems (see Chapter 6), the work here focused more specifically on the global coordination of the framework to recover from a broken TCP connection.

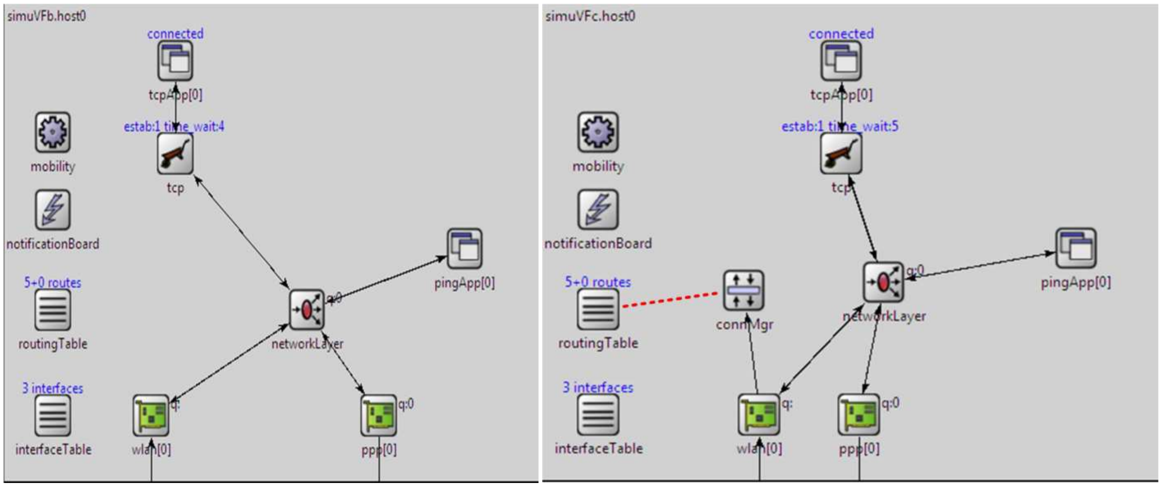
5.1.2 Choosing Validation Tool and Method

The first choice to make was between the EURECOM emulation platform and a simulation environment. Even though testing the framework with real networking protocols and applications sounded appealing, it was not convenient to gather the necessary amount of hardware and implement all the required software, including the handover trigger. Alternatively, a simulation environment looked more suitable; it is more flexible, it can be extended as needed and it allows moving the wireless host according to random patterns, providing more reliable results for this study. Moreover, it enables the parallel development of several virtual simulation test beds.

Several network simulation environments are available on the Internet: ns-2, ns-3, OMNET++, NetSim, OPNET... and have been taken into consideration. OMNET++ [82] was chosen because it offers an open license to academia; it is able to run equally under Windows or Linux operating systems. It is a component-based, discrete event simulation system, designed for building communication network simulators, multi-processors and other distributed systems. The INET framework sub-project provides many built-in implemented protocols (IEEE 802.11, IPv4 and IPv6, TCP, UDP, RTCP and RTP ...) and applications (File Transfer, HTTP, Video, VoIP ...) that were useful for this study. Moreover, a former 802.21 model, implemented in that environment and that had been used for different research projects on mobility [83] [84], was made available by some project partners. It could not be included directly here because the simulation environment has performed a major release change since it was last used. Updating that model would have required an excessive effort. However, it provided a very good reference for the setup of the simulation. The major drawback of OMNET++ is its lack of any cellular radio implementation. The same problem was faced for the former 802.21 implementation and solved by using an always-on PPP (Point-to-Point Protocol) channel tuned with parameters (connection time, bandwidth, RTT) corresponding to the behaviour of a cellular channel (LTE in our case). The results presented in the next sections have been obtained with the latest versions available at the time of completing this study: OMNET++ 4.2.2 (released on 27/03/2012) and INET-2.0.0 (released on 07/08/2012).

5.1.3 Reference Model Setup

A reference simulation model has been setup to re-create the behaviour of an existing mobile terminal and provide results to be compared with those obtained when adding the CCF. Two types of wireless hosts were used for the reference simulations presented in Section 3.3 and are pictured in Figure 44.



a) Standard Wireless Host b) Multimode Wireless Host

Figure 44: Reference Wireless Hosts

In the Standard Wireless Host (Figure 44a), a *tcpApp* application, which can be configured as FTP or HTTP, runs above TCP, a network layer built around IP and two wireless adapters: *wlan* and *ppp* (simulating *lte*). A *pingApp* is also available to trigger the execution of the Ping test. The figure also shows the *interfaceTable* module which contains the addresses and properties of each network interface, the *routingTable* module which contains the routing rules enforced by the *networkLayer* module and a mobility module which can be configured to setup the mobility scheme adopted by the MT.

Several mobility schemes are available in INET: *stationaryMobility*, *linearMobility* and others for random mobility. The pattern selected here is the R*andomWPMobility* or Random Waypoint mobility model provided by the framework. In this model, the mobile movement is linear. For each line segment, a random destination position (distributed uniformly over the playground) and a random speed is chosen. The speed is specified as a random function, expressed as *normal(meanSpeed, stddev)*, where *meanSpeed* is the average mobile speed and *stddev* the standard deviation around this value, and truncated to non-negative values. When the node reaches the target position, it waits for the time *waitTime*. After this time, the algorithm calculates a new random position, etc. For this simulation, the parameter values adopted are listed in Table 9. The interest for this model lies more on the random distribution of the MT positions, which allows triggering environment changes during the simulation, than on its random speed pattern.

Parameter	**Value**
meanSpeed	8 m/s
stddev	4m/s
waitTime	1s

Table 9: Random mobility parameters

The speed varies between 14.4 and 43.2 km/h, which is the speed of a slow car and corresponds to the scenarios described in Section 3.1. The speed of a rapid human user walk would be around 7 km/h, i.e. 2 m/s and thus much more favourable for this study, so it is not considered for the simulations.

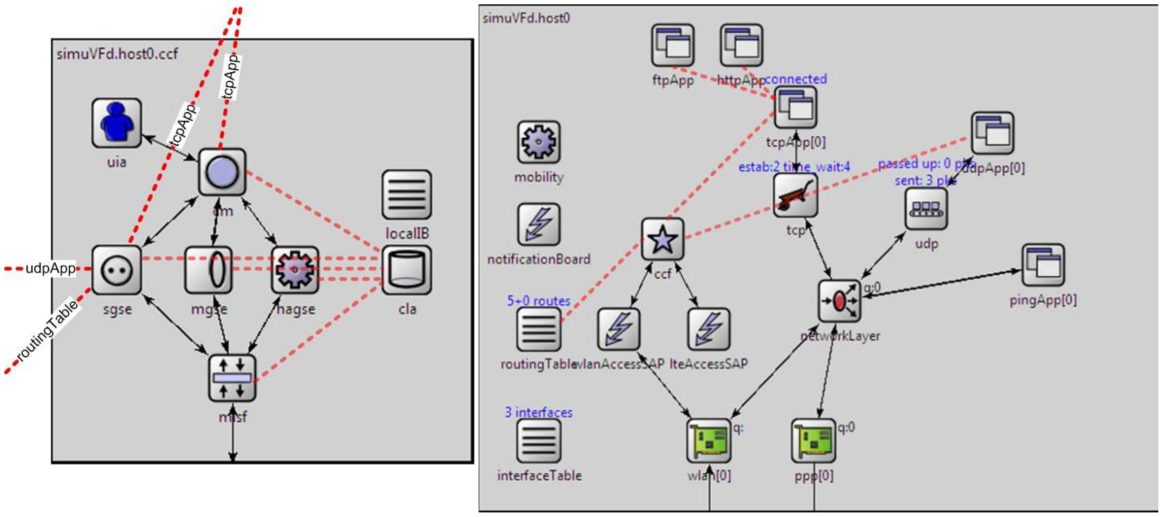
A few changes in the framework were necessary to obtain a behaviour closer to a real terminal, as observed in Section 2.1.

- The 802.11 drops silently a frame it cannot transmit on the radio instead of triggering a simulation system error. Another feature has been added here as well to dynamically retrieve the values of the SINR and the received signal power from the Link Interface.
- The number of TCP time-outs has been reduced from 12 to 3 to trigger the break of the TCP connection in an amount of time close to real life. Similarly, the congestion control mechanism has been reduced to a standard TCP implementation and TCP failure notifications activated through the socket API.
- The TCP client applications (file transfer and web client), which provided only very basic characteristics, have been reworked to be able to receive and take into account the TCP connection failure notifications. In a second step, they were adapted to support the virtual socket interface of the CCF.
- From a general point of view, several parameters have been included at wireless and application levels to collect the statistics necessary for this evaluation.

The Multimode Wireless Host (Figure 44b) implements an additional Connection Manager component, the *connMgr*, which monitors the WLAN signal level and connects to this access whenever possible, updating the internal *routingTable*. It has been designed to model the behaviour of current smartphones.

5.1.4 Implementation of the CCF Components

The implementation of the CCF internal architecture is shown in Figure 45a, and corresponds to the CCF internal definition shown in Figure 41 of Section 4.3. The picture is completed with a snapshot of the CCF Wireless Host in Figure 45b, which shows the additional modules involved in the operation of the CCF external interfaces. The arrow lines between the modules show the direct primitive and user data transfer interfaces, while the red dotted lines depict the direct function calls between the modules.



a) CCF framework b) CCF Wireless Host

Figure 45: Implementation of the CCF for the simulation

The *ccf* compound module follows the architecture described in Chapter 4. The *cm* is built with a Finite State Machine which stable states are “*Idle*”, “*C-Connected*” (i.e. control plane ready) and “*App_running*”. It moves out of the “Idle” state when the system starts (as in scenes 1a and 1b), to navigate through transient states towards the “C-Connected” state. When an application starts, the system navigates towards the “*App_running*” state, as in scenes 2 and 3. In case it receives an event triggering a handover, it navigates from “*App_running*” back to the same state once the handover is complete, as in scenes 4, 5 and 6. The *nagse* module executes the discovery, availability check and selection of the access network. The *mgse* executes the attachment to the new access and performs an update of the *routingTable* when the new connection is ready. Only one network access can be used at a given time, due to a limitation of the TCP sockets in the simulator, which also prevents experimenting the scene 6 defined in Section 4.1.3. The SGSE is shared between the *sgse* module for its control part and a set of methods in the *tcpApp* application for the virtual and real sockets operation. The *misf* module is connected to the Link Interfaces and transfers the MIS primitives back and forth in the system according to their origin, destination, types and sub-types. The *cla* module offers to the other modules a set of public methods to access its internal LIB. The framework is completed by the *uia* which interacts with the *cm* to provide user preferences at start-up and uniformly accept the *nagse* decisions.

Two Link Interfaces have been implemented and are shown in Figure 45b. The *lteAccessSAP* simulates the main RRC (Radio Resource Control) connection procedures: System Information reception, Connection Request and Setup, Reconfiguration, Measurement Control and Report. So far, it is not linked with the *ppp* interface, which simulates the LTE data plane with an always-on link. However, the link is used only when the LTE attachment or resource establishment procedure is complete and the *routingTable* module has been updated accordingly. It is tuned in the same manner as in [83], but with LTE performance figures, instead of the 3G used a few years ago. The *wlanAccessSAP* is linked with the built-in WLAN access and monitors the link attachment through the WLAN agent. The link quality is obtained by retrieving the SNR and received power measured from the radio. When they reach a threshold, the *wlanAccessSAP* triggers *linkup* or *linkdown* events to the *misf* module. The implementation of a third Link Interface module, with a *sensorSAP* simulating the arrival at the user home has been left for further study.

On the upper side, the virtual socket interface between the CCF and the application is pictured as the *tcpApp* in Figure 45b. It provides an API with function calls to the *ftpApp* and *httpApp* modules, and directly interfaces the *sgse* to implement the real socket control functions. It also transfers the session, transparently from the application, from the old socket to the new one when the network environment is changed. The *udpApp* is a module part of the *sgse* which sends a small UDP packet, containing the application PA and related event, to the application server every time the DTN is activated or a new real TCP socket has been established.

The file transfer application, *ftpApp*, is simulated as an application which periodically sends requests to the server (uplink traffic) and receives back an echo packet with a larger size (downlink traffic), maintaining the same TCP connection over the whole session. The web browsing application, *httpApp*, is simulated as a client-server application which opens a TCP connection when it starts a new session, sends a request (uplink traffic), waits until the reply arrives (downlink traffic), plus a small *thinkTime*, before it sends the next request. It closes the TCP connection after a few exchanges. After a short *reconnectInterval* amount of time, it starts another session with the server. In order to stay compliant with the objective of this thesis, the Ping test is performed from the mobile to the application server (uplink direction), in order to test the availability of the terminal connectivity.

To enable the evaluation of the benefits of the CCF operation, the following statistics are collected during each simulation run. When relevant, they are also collected in the reference scenarios.

- number of bytes transmitted and received by the applications,
- number of TCP connections opened / broken during the test,
- number of handovers 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,
- number of Echo Request packets sent and Echo Reply packets received during the Ping test.

In summary, all the components of the CCF framework have been implemented and tested, even if some of the features were streamlined due to the wide range of services defined in Chapter 4.

5.2 Simulation Execution

This section describes the execution of the simulation experiment, including the parameters used for the testing environment and the results obtained with the different applications.

5.2.1 Test Scenarios

Two identical wireless hosts move randomly in a reference network scenario pictured in Figure 46, identical to Figure 21 in Section 3.3, which includes two WLAN cells, modelled by WLAN APs and their ARs, two LTE network accesses modelled by LTE access routers, and the "*net*" Router which models the Internet network. Across the Internet is the "*srv*" Host which plays the role of an FTP or web application server to the wireless host applications. This setup allows experimenting the roaming between heterogeneous networks which makes the basis of the scenarios defined in Section 3.1. This test corresponds to the scenes 1 to 4 described in Section 4.1.3.

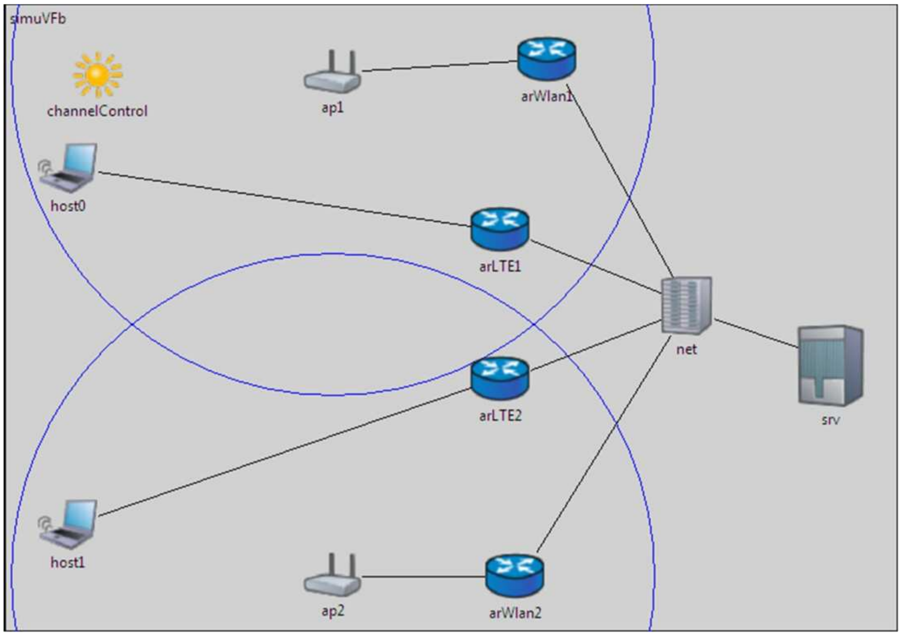


Figure 46: Simulation network setup

The tests have been performed using the two access technologies and one of the three applications available in each mobile terminal. During each simulation, both wireless hosts execute the same application. In order to obtain comparable results for all the simulation runs, some parameters have been fixed throughout all the tests: the simulation duration (2000s, i.e. 33 mn 20s), the application execution time (1500s for *ftpApp* and *httpApp*, 1800s for the Ping test) and the transmission pattern of each application. As explained in Section 5.1.3, the varying factor is the random mobility pattern applied to each terminal individually, which triggers randomly the changes in the mobile environment and the necessity to adapt the mobile behavior to those changes. Fifty simulation runs were executed to test each use case. This number, while allowing reasonable simulation time, provides a good level of confidence in the aggregation of the results obtained. In most cases, having two MTs in the same simulation run doubles the number of results obtained on a specific metric.

The LTE access network has been simulated with a *ppp* link, using the same solution as the one adopted by the former 802.21 model described in [83]. The transmission parameters used are given in Table 10. They correspond to the performance results observed in first LTE trials [85]. The WLAN access network has been simulated with the IEEE 802.11b model provided by OMNET++. The transmit power and sensibility used by the MTs are the default values provided by OMNET++ sample tests, while the transmit power and sensibility of the AP have been tuned to obtain the coverage shown in Figure 46. The two APs transmit on different channels and each mobile is configured to listen to a specific AP. Intra-technology handover has been left for a future extension of the simulation model. Table

10 also provides the parameters used for the Internet connections, simulated as Ethernet links.

Link	Data rate	Delay
srv <-> net	100 Mbps	10 ms
ar <-> net	100 Mbps	10 ms
LTE uplink	5 Mbps	20 ms
LTE downlink	10 Mbps	20 ms

Table 10: Parameters of the links in the simulated network

5.2.2 Test Results

This section presents the results obtained when adding the CCF sub-system to the MT. They take up the results obtained in Section 3.3 with two representative use cases: stationary WLAN and the insertion of a CMgr. The results obtained with the CCF are then compared with those of these two cases. This is summarized in Table 11. The selected metrics have been compiled using Matlab to provide aggregated results.

Use case Name	Scenario
WLAN 1	Stationary MT equipped with WLAN
CMGR	Moving MT including a standard CMgr
CCF	Moving MT including the CCF

Table 11: Use cases considered in the CCF evaluation

Figure 47 shows the results obtained with the Ping test. When the CCF replaces the CMgr, the loss rate drops to 0%, similar to the WLAN1 use case. This is due to the capability brought by the MISF to report a vanishing network and thus to transfer the connectivity to another link available before the older one is broken.

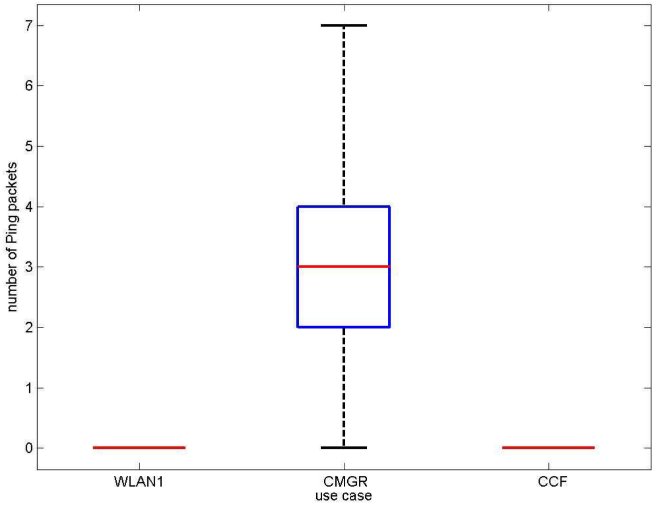


Figure 47: Ping Echo Reply packets lost during the Ping Test

A few statistics obtained with the various sets of results are presented in Table 12. These statistics are calculated as average values of the metrics measured in two different cases, *host0* or *host1* executing the file transfer application and *host0* or *host1* executing the web browsing application. The first metric is the average number of handovers. It is at the same level as the number obtained with the CMgr, yet lower due to the better control of the network access selection. The next metric is the average connection time to each technology. Because the connectivity to the WLAN network is transferred to the LTE access network before it is lost, the distribution is now more in favour of the LTE access. The next results show the benefit of the DTN-like buffering queue in the MT. This queue is used only when the application wants to transmit packets while a handover is being executed. It is emptied when the situation is back to normal. We can see that the total amount of packets queued is non-negligible in both cases, while the time during which the queue is used is very short. The same applies to the minimum and maximum number of packets queued at each handover.

Finally, the table shows the average time for executing a handover in the simulation. It is of the order of magnitude of a few hundreds of milliseconds. The average time between two handovers, for its part, shows that, compared with the test conditions, it is still large and the Ping-Pong effect is reduced.

Parameter	FTP	HTTP
Number of HO	15.65	15.47
Wlan Time	384.33	349.48
Lte Time	1366.92	1409.07
Total packets queued - max	36.00	10.00
Total packets queued - mean	16.61	3.90
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
Buffering duration	0.52	0.50
Average HO duration	0.34	0.34
Average time between HO	499.75	510.42

Table 12: Measures for buffering mechanism

Figure 48 shows the average amount of traffic transferred when testing with the web browsing application. The amount of traffic is slightly lower than in the CMGR use case. A larger part of the traffic is transferred through the LTE access which offers lower bandwidth and throughput capability. Because of the interactive nature of the application which waits until the reply arrives before sending the next request, this has an impact here on the overall performance

.

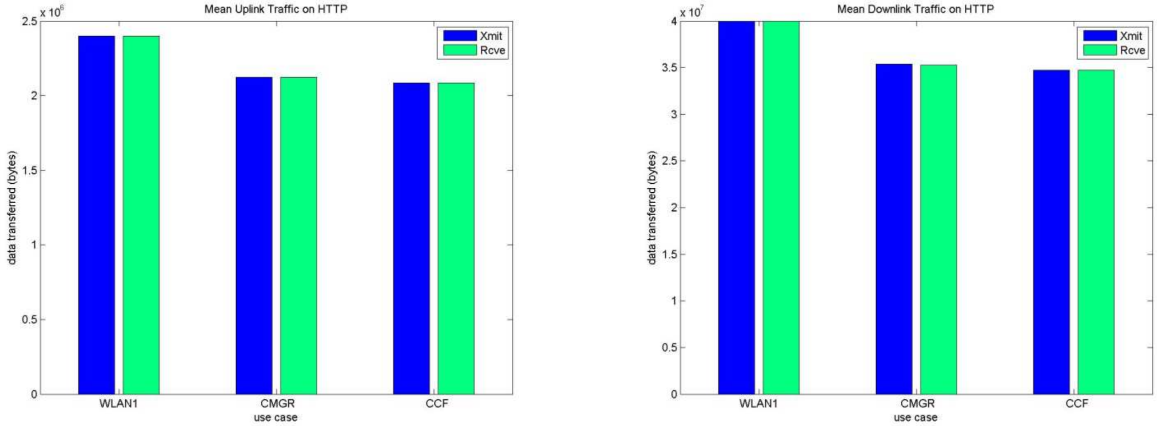


Figure 48: Comparison of traffic during a HTTP session for uplink (left) and downlink (right)

Figure 49 shows that, when using the CCF, the number of broken session (–*BRK*) is reduced to zero. All the sessions end successfully. A very small amount of requests are retried by the application because the downlink packets were lost during the handover (between 0 and 7 maximum for the two mobiles). This compensates the small decrease of traffic performance.

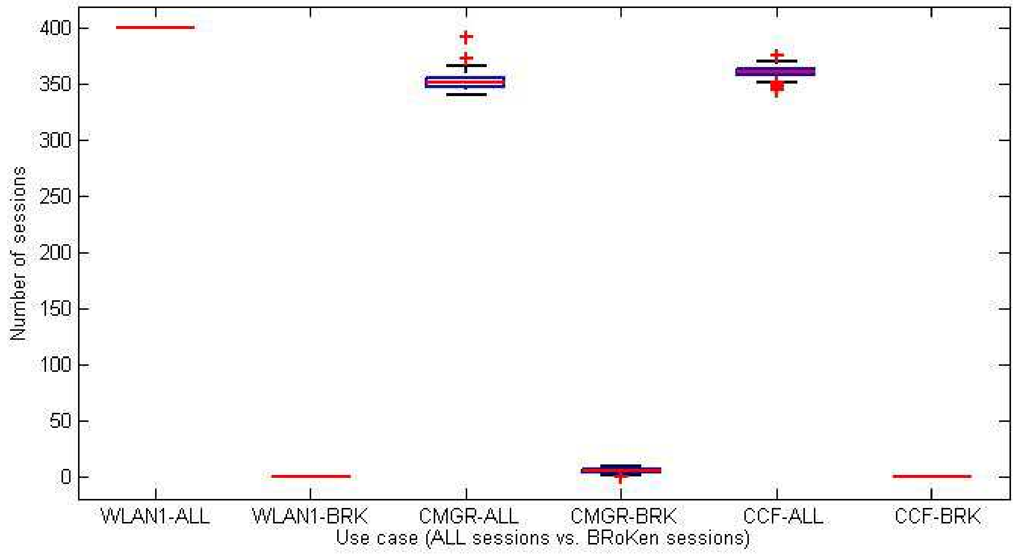


Figure 49: number of HTTP sessions broken vs. total number of sessions

Figure 50a (left) shows the average amount of uplink FTP traffic transferred in all three use cases. In the CMGR use case, the session was broken and could not be recovered. Thanks to the virtual socket interface, the re-establishment of a fresh new TCP connection and the buffering queue, the same amount of traffic is transferred without any loss. Figure 50b (right) shows almost the same results for the downlink traffic.

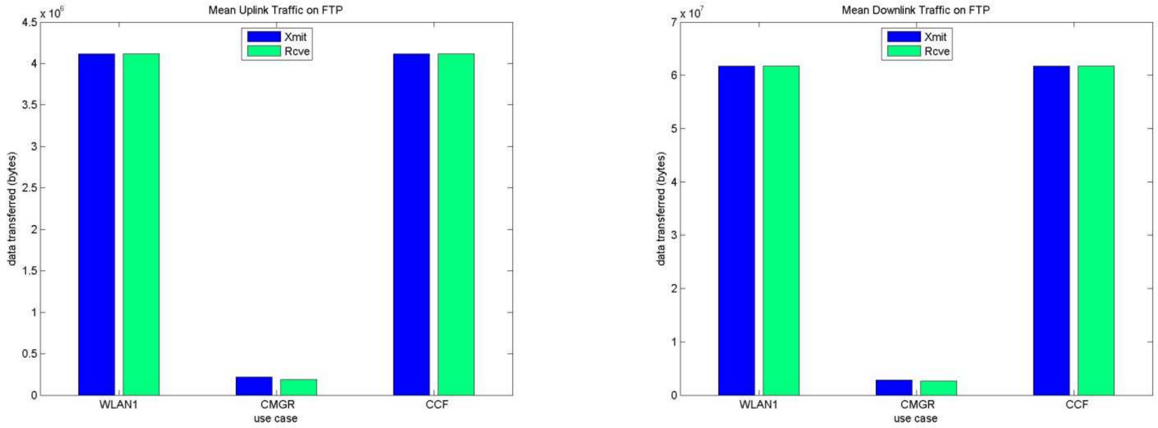


Figure 50: Comparison of traffic during a FTP session for uplink (left) and downlink (right)

Figure 51 and Figure 52 show the time when the last packet of the file transfer session was received by the server and the mobile host. This result confirms the conclusion of Figure 50. At the contrary of the CMGR use case, the file transfer application has not failed and could complete its execution thanks to the *sgse*, as if in a stationary terminal and totally unaffected by the large number of heterogeneous access network changes executed.

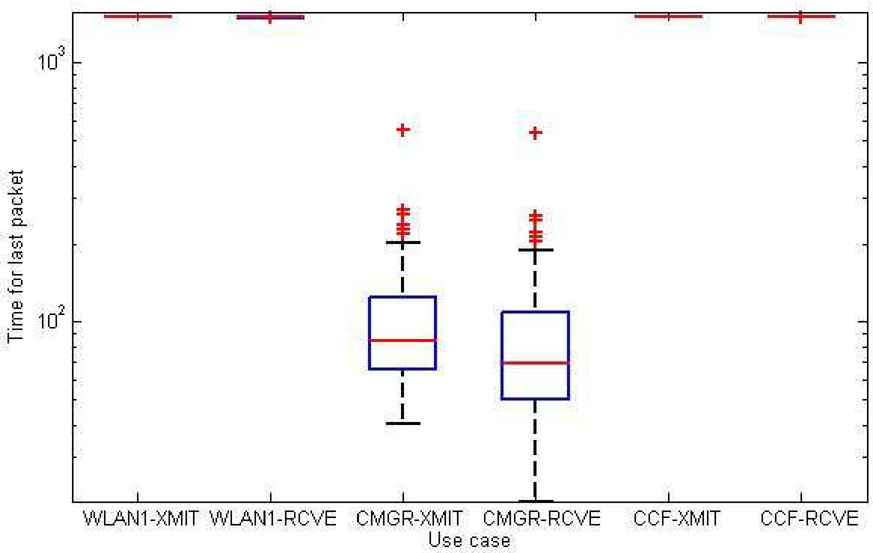


Figure 51: Last packet transmission and reception time at mobile terminal

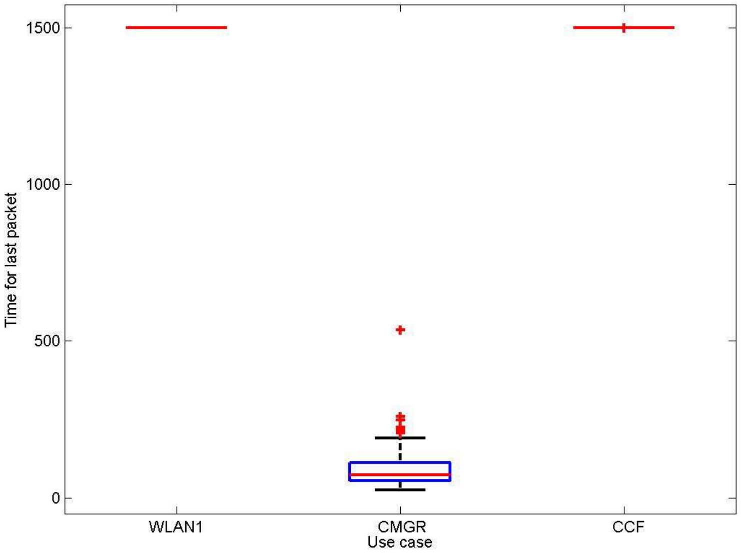


Figure 52: Last packet reception time at application server

In order to evaluate the impact of the CCF on the terminal processing time, a sample typical run has been conducted to measure the amount of discrete events involving the *ccf*, the *wlanAccessSAP* and the *lteAccessSAP* components, vs. the total number of events for this simulation run. Table 13 shows the measured values for a test duration of 350s, reduced due to limitations of the framework traces. For the same reason, a specific test protocol is executed here, during which one MT executes the file transfer application and performs 12 handovers.

	Number of events executed	Percentage
Total of the simulation	761839	
ccf contribution	409	0.0537%
wlanAccessSAP contribution	774	0.1016%
lteAccessSAP contribution	50	0.0066%

Table 13: Measured processing time

This result must be taken cautiously, because it results from a simulation and not a real life execution. Moreover, it considers each discrete event as one entry and not the amount of processing induced by that event. However, it shows that the most important contribution comes from the *wlanAccessSAP* which retrieves periodically the received signal level to test the status of the link. Since it is executed at the edge of the framework, it does not have an impact on the CCF contribution. This contribution is only due to reaction to events and thus remains very low. From a global perspective, the total impact in the case of 12 handovers, which is a large number considering the scenarios presented in Section 3.1, is under 0.2% of the total processing power consumed by the mobile operation, so it can be considered as very low.

5.3 Summary

This chapter has presented the result of the CCF prototype development and compared it with two other cases: standard multimode terminal with no connectivity control and mobile terminal equipped with a standard Connection Manager.

The prototype has been developed as a simulation running under the OMNET++ tool. This simulation executes a file transfer or a web browsing application in a mobile terminal under one of the three different configurations described above. The standard terminal is stationary while the other two move randomly across a playground containing an always-on LTE access and a WLAN cell with a limited coverage.

The implementation of the CCF sub-system for the OMNET++ environment is described. The choices made for the different modules, including the Link interfaces, the virtual socket interface and the metrics collected are presented. This section is completed by a description of the parameters selected for the execution of the simulation.

The results obtained from the simulation runs are aggregated and analysed in Table 14, which can be put in parallel to Table 6 in Section 3.3. They show that when executing a Ping test, no packet is lost thanks to the CCF connectivity handling. When executing a file transfer application, the virtual TCP connection is not broken. The application can complete its task until the end and transfer all its packets. Some of them require the additional support of the DTN-like buffering technique. However their number during a specific handover remains very low, at a level quite acceptable compared to the amount of memory available in a mobile terminal. When executing the web browsing application, no session is broken anymore midway to its completion. The amount of traffic transferred is slightly reduced compared to the case with the CMgr, because a larger part of the data traffic is transferred on the safer LTE access which, on the other hand, offers a reduced bandwidth. Finally, some statistics are collected which demonstrate the low fingerprint of the CCF operation on the MT processing power and the short time required to transfer from one access network to the next.

Use case Name	Ping test	FTP test	HTTP test
WLAN 1	No loss	No loss	No loss
CMGR	A few packets lost	Loss without recovery	Loss with recovery – Lower traffic rate – A few sessions broken.
CCF	No loss	No loss	No loss – Lower traffic rate – No session broken.

Table 14: Traffic observations according to the application

Due to the wide functional coverage of the CCF and the potentially large number of use cases that could have been deployed to test it, the work here focused only on the more critical issue of the TCP recovery. A few other features were streamlined and left for future work. This model could be extended to support Link Interfaces controlling the terminal power supply and a sensor detecting the presence at the user's home for instance, as described in Section 4.3.2. Another extension would be to execute more complex tests where the user speed or the application data rate would be variable. Yet another extension would consider a UDP-based application (e.g. pure VoIP or video streaming) to verify that it is more resilient to network changes, but still performs better with the CCF.

CHAPTER 6 - APPLICATION TO REAL SYSTEMS AND PROJECTS

The architecture and concepts of the proposed framework has been presented in Chapter 4. It is based on three main principles: *(i)* the insertion of an abstraction layer between the managed resources (wireless access or other embedded device) and the network services, *(ii)* the provision and coordination of generic service enablers to care for specialized connectivity tasks, and finally, *(iii)* the sharing of knowledge across the whole framework. Part of these principles have been implemented and evaluated in design of real systems and projects involving multimode mobile terminals: the integration of Media Independent Services in a beyond-3G (B3G) experimental platform, geo-broadcasting for intelligent transport systems communications, communication technology selection and Management layer design for an ITS Station, wireless access abstraction supporting mobile video applications and finally, mapping at the lower interface level between Media Independent primitives and EPS/LTE procedures. This chapter describes how the CCF principles are applied to these systems, highlighting its impact, their large variety and the different situations where it brings benefits and efficiency.

6.1 Seamless Mobility for the Integration of a Beyond-3G Cellular Access

The first application corresponds to the first use case in Section 3.1 and targets the integration of a cellular access in a seamless localized mobility framework. To cope with the heterogeneity of the technologies it addresses, the system designed requires an abstracted framework for mobility management and control. The IEEE 802.21 standard has been applied and evolved to enable the integration of a B3G cellular access, very close to the LTE specifications of the 3GPP, with the objective to improve the global control operations between heterogeneous technologies. It results in the integration of a selection of MIH primitives with the cellular air interface procedures available in a B3G experimental platform, for Mobility, but also QoS resource allocation and multicast support (MBMS). This integration has been implemented and evaluated in a real-time environment with several other types of technologies, namely WLAN and DVB, for the DAIDALOS project [86] and published in [2]. For this study, the cellular or evolved-UMTS access was provided by EURECOM’s experimental software radio platform “wireless3G4free” [87] which features a direct-interconnection between the IP protocol and the UMTS air interface. One of the key objectives of this work was the close integration of Mobility and QoS signalling for the various technologies [88]. One of the issues, specific to the cellular air interface, was the fact that it is conceptually very different from the 802.x access, in the sense that it is a connection-oriented technology and not a shared medium, and that it has full awareness and impact of the QoS.

As shown in Figure 53, in the DAIDALOS implementation, the abstraction model is split into two main sub-layers. The MIHF is common to all the access technologies and harmonizes the radio control of the wireless accesses. Each radio technology is controlled by a specific module, the Radio Access Layer (RAL). The RAL is the component responsible for the specificities of the technology, mostly in charge of reporting the measurements and executing the interface activation and deactivation on request.

The mobility part has been implemented according to the IEEE 802.21 model, associated with the NetLMM (Network-based Localized Mobility Management) and MIPv6 mobility protocols. The measurements, taken in the UMTS Access Stratum by the RRC layer, such as the signal level, the interference level and the path loss are directly uploaded to the RAL which smoothed them out, analysed their variations and generated reports as Events to the MIHF and its users. The RAL was also able to report a Link_Up event when the RRC had been able to successfully monitor the System Information Broadcast (SIB), camp on a suitable cell and attach to this cell. When the MT executed a handover to the UMTS access network, the interface was activated by a “Link Action: Link Power Up” primitive. This triggered the network attachment of the UMTS interface to the Access Router and the establishment of an IP route between the two nodes.

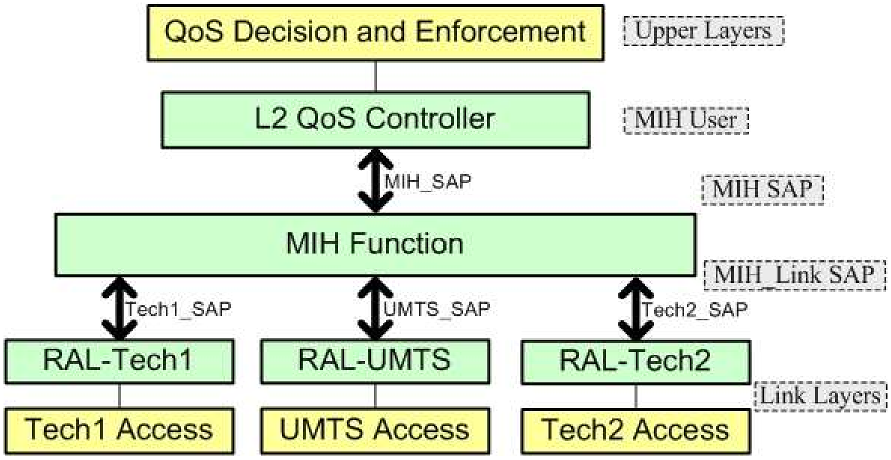


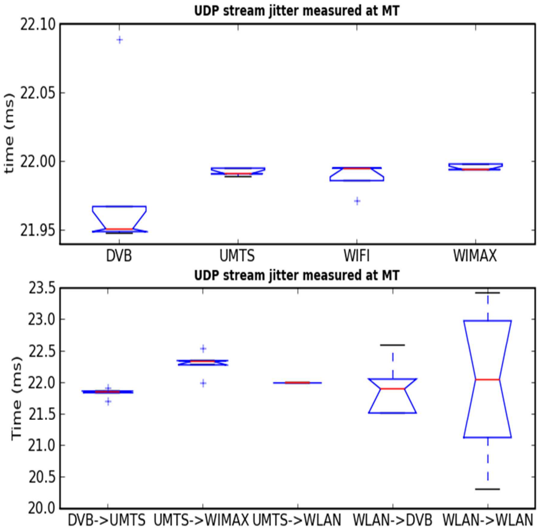
Figure 53: MIH-enabled architecture for Resources Allocation

The QoS part required a more innovative design. Since the UMTS is a connection-oriented technology, it was mandatory that some radio resources be allocated with a specific QoS to enable the transfer of the user data traffic. The support of QoS in 802.21 considers the guarantee of service continuity during the handover, taking into account parameters such as the throughput or the Class of Service for example. However, it does not consider the actual allocation of resources during the handover with the 802.21 primitives. The QoS support available in this network was performed at the Layer 3 level by centralized QoS Decision

Points and distributed QoS Enforcement Points located in the ARs. As shown in Figure 53, they played the role of upper layers. The abstract interface has been enhanced to transfer requirements from the enforcement points to the Link Layer entities. The L2 QoS Controller module, a MIH User, bridges the gap between the L3 and L2 QoS subsystems. It was able to generate any number of MIH QoS reservation primitives for each L3 QoS request received from the upper layers. This extension of the MIH architecture allowed a joint interaction between the mobility and the QoS support in the sense that the allocation of the data radio channels was performed during the handover execution phase.

When the MT committed its handover to the UMTS network, the RAL in the AR received a Link Resource Activate command, carrying the QoS parameters, such as the reserved bit rate or the Layer 3 QoS class identifier. The parameters were mapped onto one of the radio QoS classes available in the UMTS platform and the resource reservation was prepared. When the handover was executed and as soon as the MT was attached to the new AR, the RAL triggered the physical allocation of the resources in the Access Stratum. Preparing the resources reservation ahead from the actual handover execution accelerated their establishment and improved the handover performance. Since this framework was very flexible, it has also been used for the establishment and release of MBMS point-to-multipoint radio channels [75]. For a more efficient operation, it is better to establish them dynamically and not waste radio resources when none of the MT in the cell is interested in joining the multicast group. The procedure was identical to the one used for the point-to-point radio resources, except that a specific identifier in the MIH command indicates that a multicast resource was concerned. The received QoS parameters were available for the downlink direction only and mapped onto one of the specific MBMS QoS classes. The study on multicast resources and technology support has been used as a basis for the 802.21b revision of the standard.

In the validation testbed, this work has been applied to a road scenario use case where the car passenger's terminal is connected to WLAN in front of the university. He starts consuming an entertainment video. As the car moves out of the campus, the terminal handovers to DVB with UMTS as return channel, the handover being triggered by the decreasing signal strength of the WLAN. After a while, some personal preference settings related to the DVB rate fires some rules, the terminal drops the DVB connection and continues on UMTS only, followed by two additional handovers to and between the WLAN accesses. This sequence of heterogeneous handovers has been successfully implemented, integrated and demonstrated. In terms of implementation, it was much easier to develop the layers above the RAL and the MIHF, since they did not have to take care of the technology specificities, such as the UMTS technical details and parameters. Figure 54 and Figure 55 picture some of the measurements that were performed during the final tests. They show the global results for all the access technologies. However, the interest and discussion here is restricted to the scope of the UMTS interface.



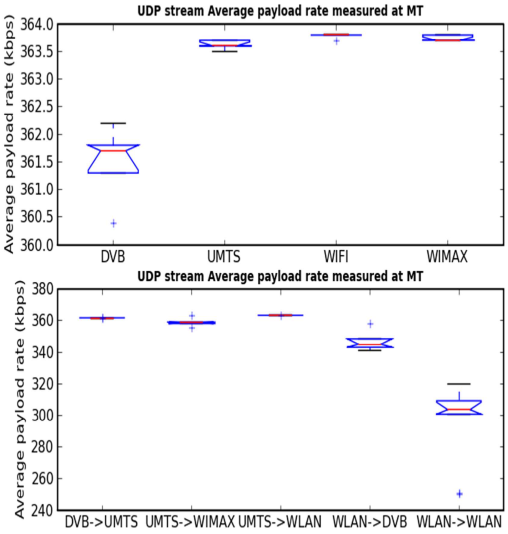


Figure 54: Jitter during sequence of handovers

Figure 55: Payload rate during sequence of handovers

A data packet stream was transferred at a rate of 384 Kbits/s, which is a typical rate for 3G systems. The performance parameters taken into account were the average delay, the packet loss, the jitter and the modification of the payload rate. When the UMTS is involved in the handover, the average handover delay has been measured at around 6 seconds, with a non-significant disruption time. This result was expected because the scenario applies to inter-technology handovers with “make-before-break” capabilities, which means that the new network attachment is performed before the old one is broken. The handover delay starts from the trigger of the handover to its final completion and involves several steps of signalling exchanges. For the same reason, it was confirmed that the packet loss was null during the handovers.

Figure 54 shows the jitter measured at the MT during each handover, compared with the jitter obtained for each technology in a stable situation. The diagram shows that the jitter is almost identical during the handover as it was in the stable situation, e.g. very close to 22 milliseconds. A similar result was obtained when measuring the packet delay during the handover. Figure 55 shows the average payload rate during steady traffic and then during the handovers. Again, we can observe here that the payload rate is kept at the same level during the handovers involving the UMTS interface. We can deduce that there is no impact from the dynamic radio resource reservation and establishment. Thanks to this integrated mobility and QoS architecture which allows preparing for the handover beforehand, all the radio resources could be established very early, as soon as the handover took place. As a result, it was possible to execute successfully a sequence of seamless inter-technology handovers between three different types of access, involving cellular networks, with a low perturbation of the jitter or packet delay.

The application case presented in this section has described how the MIS based on the IEEE 802.21 have been extended to support QoS resource allocation in addition to seamless mobility, targeting the integration of a beyond-3G cellular access. In this system, the upper layers were directly the Mobility management and the QoS controller entities, showcasing the flexibility of the MISF abstraction.

6.2 Intelligent Transport Systems Communications for Cross-Layer Geo-Broadcasting

The target of mobile communications is sometimes a limited geographical area. Geographical networking (or geo-casting) is currently being defined, with proposals to indicate the target area by adding a geographical enhancement to the IP protocol [89]. In the domain of ITS communications, several use cases imply the distribution of geographically constrained notifications to the ITSS, either vehicle stations or user personal handheld devices. They transmit periodic permanent or temporary messages broadcasted from an infrastructure centre. This application study corresponds to the second scenario of Section 3.1. The messages may contain road warnings to inform the driver of a potential hazard, contextual speed limit information or regulations, information about road modifications due to changes in the traffic flow or entertainment, and practical information for the user. In order to reach the maximum number of mobile users and terminals, including the personal handhelds, some traffic and safety applications can become more efficient if their data are transferred in more than one access network, allowing the usage of multiple technologies in parallel and the selection of the best and most convenient one by each user or terminal.

Moreover, we have proposed [3] to complete the solution based on the IP protocol by defining some cross-layer cooperation and an abstraction framework where Network and Link layers contribute to improve the efficiency of the geographical coverage. It is described using the MBMS and the multicast/broadcast functional entities standardised in the 3GPP for the cellular networks, adding a geographical selection based on network topology at the head of the mobile network broadcast distribution. But it is generic enough to be applied to other types of access networks under similar conditions.

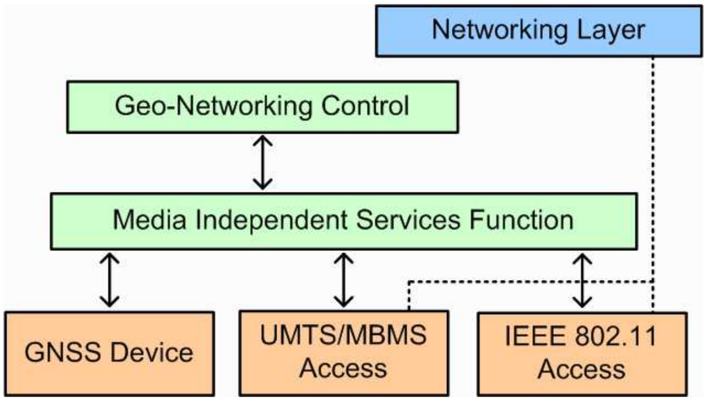


Figure 56: Cross-layer Geo-Broadcasting

The proposed abstraction framework is illustrated in Figure 56. When a broadcast session has been successfully started in the network, data packets are transmitted on the established data bearers. A mobile terminal, which has indicated interest in the reception of the service, enters in the covered area, probably larger than the target area, and starts receiving these packets through its active technology. The initial packets received by the wireless interface are forwarded by the abstraction or Media Independent framework towards a GeoNetworking Control entity (GNC) which acts here as a geo-casting GSE. The same abstraction mechanism is used to report the MT dynamic geographical coordinates to the

GNC. These coordinates are often provided by a GNSS device, such as a GPS (Global Positioning System), but can also be retrieved from the localization functionality of another technology, hence the benefits of using an abstraction layer. The GNC computes an initial correlation between the measured coordinates and the target area information contained in the network header of the IP packet. If a match is found, the decision is taken to continue the reception of the broadcast session on the MBMS channel. The flow is transferred to the corresponding Network layer. If no match has been found, the decision is taken to filter out the broadcast session. De-registration from the service is not appropriate here because it would mean completely stopping the reception of this type of information. A command is sent through the MISF to the link access layer and the MBMS Bearer Service in the MT, which tunes off the technology for this flow and thus avoids a pointless filtering further up in the Network layer. Later on, a periodic re-evaluation of the correlation is performed with an updated input from the localization system. It takes into account the MT mobility, potentially starts over the reception of the broadcast information which is then forwarded upwards to the Network layer. The correlation periodic re-evaluation is stopped when the flow of packets related to this service dries up, either because the terminal has moved out of the broadcast area or because the traffic information itself is not relevant anymore, and the broadcasting has been stopped.

By using this process, the MBMS at the Link layer access cooperates with the Network layer to improve the whole system efficiency and enhance its capability. The cooperation between the two layers enables a more precise and efficient delivery of the I2V broadcast information. When the MBMS is used, this system takes advantage of the comprehensive knowledge of the network topology by the UMTS Core Network and of its broadcasting services. It benefits from an already deployed infrastructure which can easily cover large or small areas, as needed. This framework has also been proposed to disseminate an emergency public warning notification in a geographically constrained zone for the RATCOM project [1] [4].

The application case presented in this section has shown how the abstraction of heterogeneous network interfaces and of other hardware devices, associated to a cross-layer design, could enhance the geo-broadcasting of public messages for ITS or warning notifications. Moreover, the concept of generic service enabler has been exploited to control the geo-casting of the messages.

6.3 Access Selection and ITS Management in Vehicular Communications

The domain of vehicular communications is often very specific and requires some adaptation of the existing technologies. Section 2.6 has presented the layered architecture of an ITSS and Section 3.1, a scenario showcasing this type of communications. In this domain as well, the introduction of cross-layer mechanisms is required and beneficial. This applies first to the management of identities in the ITSS, taking into account very strict constraints in terms of privacy [5]. In order to be able to communicate with one another, the ITSS must be identifiable in the network and at each protocol layer of the architecture. It is thus needed to define a global set of identities which satisfy these constraints. Some identifiers may be reused from the technologies involved in the communication stack while others are ITS-specific. Due to the safety applications, the identifiers of an ITSS must be unique in order to individually recognize the station during its communications with peer entities in the network. Moreover, they must be changed periodically due to the privacy issue. One possible approach that is applied in the FOTs is to use a single short-term identity or “pseudonym”, handled globally, and used to further determine the identifiers required for the operation of the different layers of the model. In summary, the required identifiers must be coordinated across the various layers to simplify, strengthen and streamline the transmission of packets over the different accesses, while keeping the communications secure and reliable. These mechanisms are controlled by the Management layer, a vertical cross-layer component of the ITSS, highlighted in Figure 57, which coordinates the operation of the horizontal layers: Access Technologies, the Network and Transport layer, the Facilities layer and the Application.

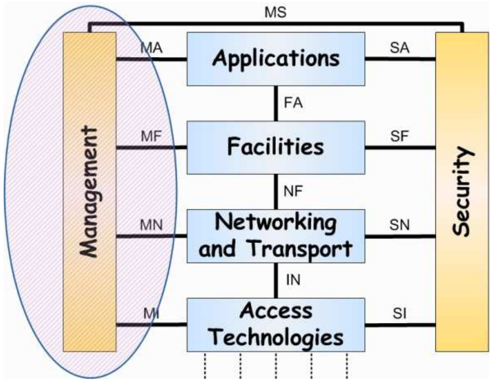


Figure 57: Management layer in the ITS Station model

Also located in the Management layer is the communication technology selection which decides of the most suitable set of communication protocols and access technologies to carry the messages of a given ITS cooperative application. As shown in Section 2.4, network access selection is generally performed at flow, or even device level. A specificity of ITS communications is that this decision must be made dynamically, according to the type of the message to transmit and closely related with the application which indicates its own requirements and the current context of the ITSS.

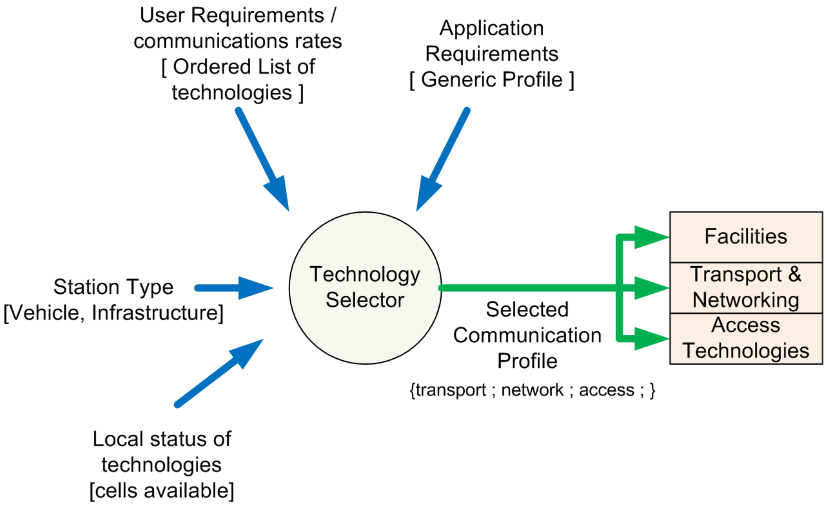

User Requirements / communications rates [Ordered List of technologies]
Application Requirements [Generic Profile]
Station Type [Vehicle, Infrastructure]
Technology Selector
Selected Communication Profile
{transport ; network ; access ; }
Facilities
Transport & Networking
Access Technologies
Local status of technologies [cells available]

Figure 58: ITS communications technology selection

A study performed for the iTetris traffic simulation platform [90] has identified a minimal set of inputs to make this decision. The global process is pictured in Figure 58. The first input consists in the type of the originating and destination ITSS. For example, it may be a vehicle, a RSU (Road Side Unit) or a central traffic station. The second set of inputs is user-originated, especially in the case of a mobile ITSS. Based on user preferences and subscription rates, the Management layer builds dynamically an ordered list of access networks. This list may be updated at any time if one of the conditions used to build it has changed. In the case of an Infrastructure-originated transmission, this list is built by matching the destination target area description with the list of transmitting stations (base stations or RSUs) covering that area. The Management layer receives from the application the profile requirement associated to the message type. The same type of message (e.g., event notification) may have different values for this field under different conditions or applications. For example, a profile may look like "high emergency message, vehicle to vehicle only". A set of these profiles is pre-defined in the ITSS Management database and the message-to-profile association is determined by the application. The fourth and last input is built from the ITSS context and contains the current status of the network accesses, observed locally by the ITSS, with parameters such as signal quality, radio coverage, network load or distance to the destination station. The objective of the technology decision is to choose the most suitable communication profile which includes the following information: Transport protocol, Network protocol, Access technology and channel (when needed).

This approach has been applied to the implementation of the communication technology selector in the platform and has led to the definition of two procedures, the first one applied to the vehicle decision process, shown in Figure 59, and a second one applied to the global infrastructure (central station) decision process. Based on "Requirements Generic Profiles" (GP1, GP2... in the figure) passed by the iTetris applications, a communication profile (CP1, CP2...) is selected, taking into account the current availability of the technologies and the user preferences. These processes have been implemented and evaluated with a positive feedback, as documented in [6]. The overall solution has also been proposed to the ETSI ITS Technical Committee for the standardization of the ITSS Management layer [91].

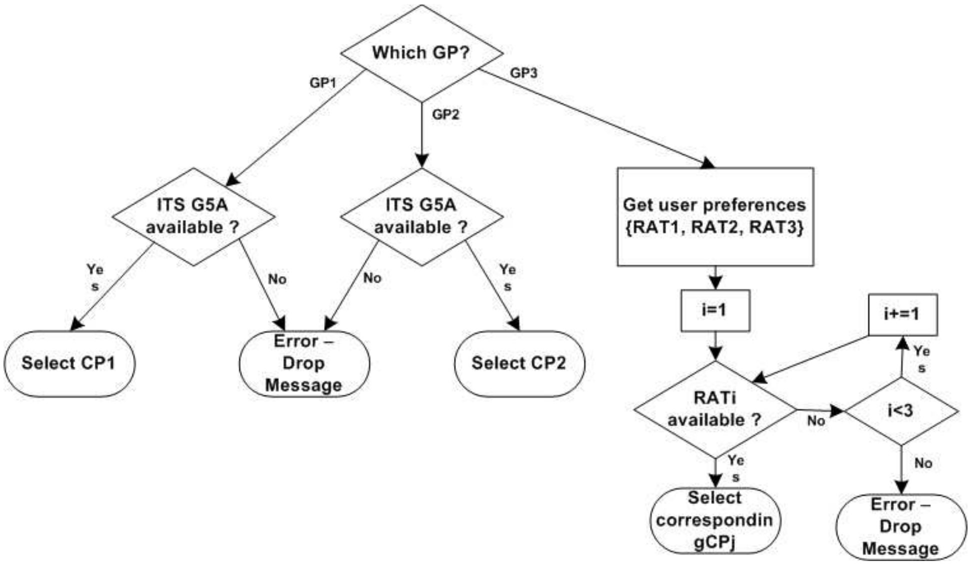


Figure 59: Communication profile selection algorithm in a vehicle

This concept has been extended to the specification of the functional architecture of the complete Management layer for the ITS Station model (see Section 2.6) in the SCORE@F project [7]. The global picture of the designed entity is shown in Figure 60. In addition to some ITS specific entities, others more generic ones could be identified. The first one is the internal station management, which handles the initialization and updated configuration of the whole system. It keeps an up-to-date status of the system components and their current properties. Next is the service management which distributes service announcements and forwards the received information to the relevant applications. An important entity is the communications management which controls the communication profile selection that was described above, the channels choice and the setup of the routing protocols tables. Last but not least is the cross-layer management which handles the sharing of parameters common to multiple layers and the cross-layer notifications. These modules are completed by a virtual database module, the Management Information Base (MIB) which aggregates all the system data, stores them and provides them on request when needed.

This design has been simplified for its implementation in the FOT. The main features chosen are the ITS station initialization, the cross-layer database dynamic management and the communication profile selection. The result is shown in Figure 61. On the same basis as in the CCF, the modules are split in two parts: the Management Core which provides the cross-layer service and a set of technology-specific Communications Modules (named FAC-CM, GN-CM, G5-CM, LTE-CM, etc.) which provide the interface between the protocols and the Management Core. This layer implements a functional concept similar to the Generic Service Enablers, especially the NAGSE which corresponds to the communication profile selection module. It maintains the shared knowledge of the system in its MIB, in the same way as the CLA does for the LIB in the CCF.

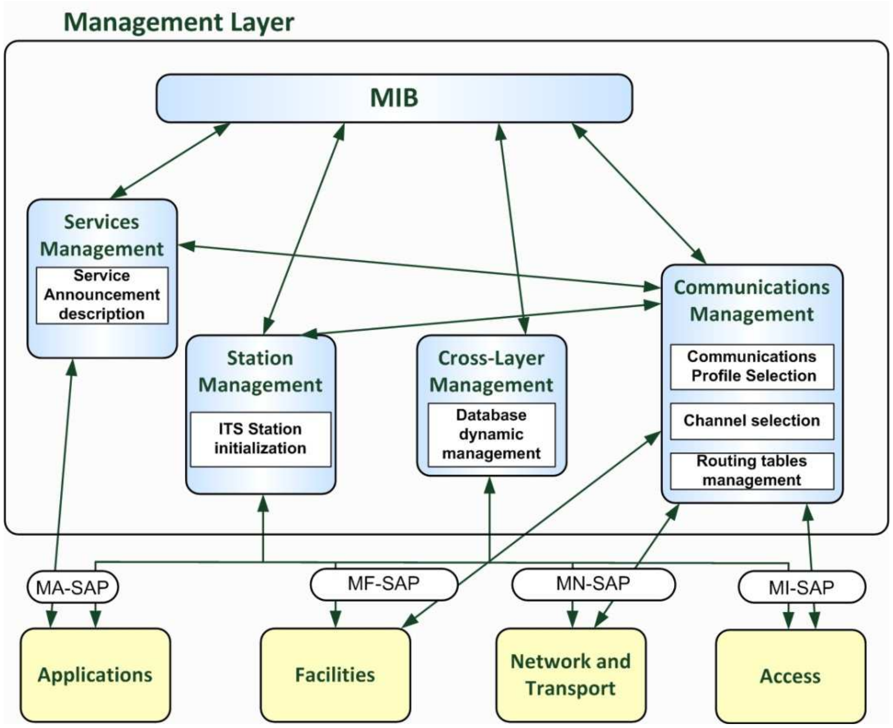


Figure 60: ITS Station Management layer design

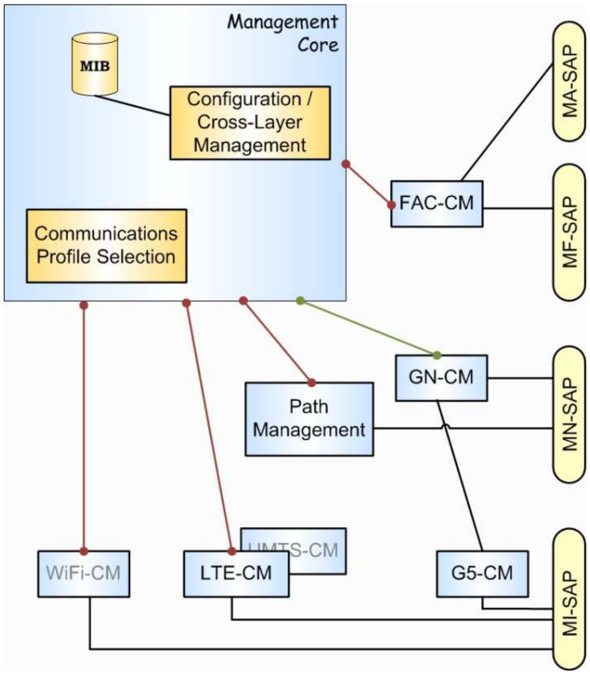


Figure 61: SCORE@F ITS Management layer

The application case presented in this section has shown the implementation of a cross-layer component responsible to collect and enable the sharing of the system parameters. As in the CCF, it aggregates these data in a local information base. Moreover, a generic service enabler has been presented. It performs the selection of the best communication profile to disseminate a specific message or flow in an ITS Station, going beyond the access network selection.

6.4 Cross-Layer Enhancement of Operated Networks for Video Services

From a general point of view, cross-layer approaches allow an improved evolution for mobile networks. This has been applied to the MEDIEVAL project [8] aiming to enhance the current mobile Internet and deliver more efficiently mobile video applications. This application study is linked with the first use case of Section 3.1 and is actually an evolution of the application of Media Independent Services described in Section 6.1 to the delivery of video data traffic. Inside its global architecture, pictured in Figure 62, this project considers a wireless abstract interface, which guarantees a transparent interaction between the underlying wireless technologies and the video-aware upper layers. A cross-layer relationship is established with the upper components, in this case the mobility management and transport optimization sub-systems, to exchange information about the capabilities of the lower components, as well as to configure them [9]. It forwards downwards the received commands which carry the setup of the wireless access, translating the generic configuration parameters received from the upper layers into specific extensions defined by each of the heterogeneous technologies. It extends the Media Independent signalling functionalities provided by the IEEE 802.21 by supporting additional features such as resource configuration or priority setting. This allows the setting of priorities to video frame and the optimization of the allocation of the radio resources, adapted specifically to the video flows it has to deliver in a certain context. It runs counter to existing systems which, rely only on the algorithms located inside the video application, and improves the user Quality of Experience (QoE). In other cases, the widespread diffusion of a similar content may target some group usage, where the cellular MBMS, for example, is a good access candidate and allows the mobile operator to manage his resources more efficiently.

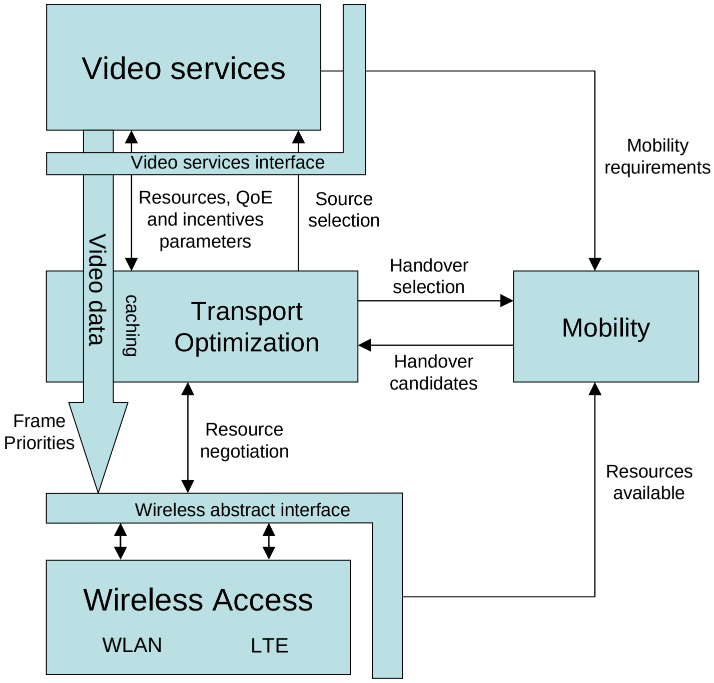


Figure 62: MEDIEVAL Functional Architecture

The objective of the Wireless Access subsystem [10] is to design novel mechanisms in the wireless technologies that optimize the video delivery in the last hop. Its main focus is on contention-based wireless accesses (e.g., WLAN or IEEE 802.11) and coordination-based wireless accesses (e.g., LTE-A or LTE-Advanced from 3GPP). It is completed by the definition of an abstract interface that hides the technology specificities to the higher layers when executing cross-layer optimizations. As a consequence, the Wireless Access is split into three main functional blocks.

- The *Layer 2.5 Abstraction component* provides the generic interfaces between video specific functions (i.e., transport, services and mobility) and wireless accesses.
- The *WLAN component* has technology-specific functions addressing all mechanisms related to the contention-based wireless accesses.
- The *LTE-Advanced component*, on the contrary, contains the optimization features for coordination-based wireless accesses.

Figure 63 shows a global view of the sub-system, highlighting the parts related to this study.

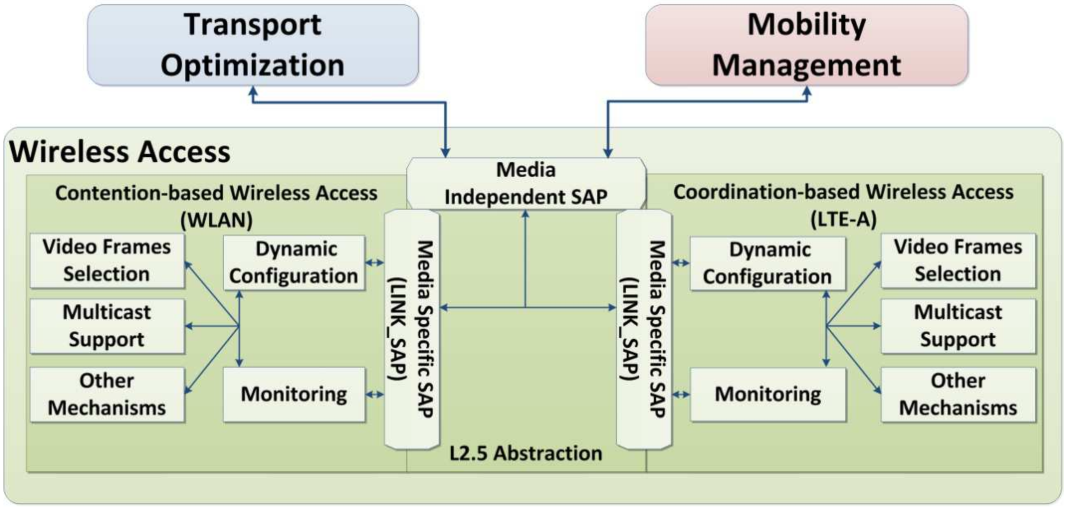


Figure 63: Global view of the Wireless Access sub-system

On top of the technology-specific entities, the project considers an *L2.5 abstraction* component. It is the heart of the video delivery optimization process. It provides the mechanisms for the interactions between the wireless technologies and the upper layers, with the objective to perform cross-layer functionalities and underlying heterogeneous technologies abstraction. Its abstract interface is designed as an extension of the standard IEEE 802.21 MIH Services and improves them by taking into account the characteristics of the video flows.

For each technology, optimizations are introduced to enhance the efficiency of the video flows delivery over the air interface, expressed as a set of common generic functions. A *Dynamic Configuration* module takes into account the requests from the upper layers and the characteristics of the video flows to setup the network interface or establish radio channels able to accommodate an upcoming data flow. It works by defining a utility function which permits to allocate resources in an optimal manner, providing the optimal set of parameters to each technology. A *Monitoring* function is responsible to dynamically retrieve the information related to the access network availability and quality, and to provide it to the upper layers through the abstract interface. Moreover, it senses its environment searching for new available networks and analyses their capacity, user registration level, bandwidth usage and remaining resources. A cross-layer *Video Frames Selection* function is able to perform rate adaptation based on the channel conditions, yet avoiding deep packet inspection. This is achieved by performing a selection of the received video frames according to a dedicated marking previously introduced in the IP packet headers. When a certain level of congestion has been detected in the network, the data packets are marked for prioritisation by the Transport Optimisation sub-system. The lower priority packets can then be dropped before the video frames are actually managed by the Link layer protocols, according to the receiver capabilities, in order to reduce the bandwidth occupation and loosen the level of traffic load in the last wireless hop. The upper layer makes the decision and marks the IP packets, based on the results of its algorithms, while the decision is executed in the wireless link interface. *Enhanced multicast mechanisms* have the objective to improve the efficiency of the eMBMS (evolved MBMS) bearer service, fully exploiting the cross-layer optimization [11]. In this design, the bearer is set-up dynamically and supports multicast mode, contrarily to the latest developments in the 3GPP LTE-A. This is made possible by the cross-layer exchange between the AR and the LTE PoA, which provides ahead of the "MBMS session" start the parameters necessary to establish the multicast session. Several levels of QoS can be configured, according to the traffic class and data rate provided by the upper layers in the resource activation primitive.

The application case presented in this section has shown the implementation and extension of the wireless access abstraction to optimize the delivery of video traffic to a mobile device on the last hop. The structure of the Link Interfaces has been developed for contention-based and coordination-based network access technologies.

6.5 Mapping of MIH Primitives to the EPS/LTE Protocols

The Evolved Packet System (EPS) is the next step of network system following the UMTS (commonly known as 3G) which brings strong enhancement and improved performance to mobile communications. The EPS and the LTE wireless access are involved in all three use cases presented in Section 3.1. The 802.21 standard contains media-specific mapping norms for the MIH_LINK_SAP using existing functionality provided by the addressed access technologies. However, for cellular technologies, only 3GPP-UMTS and 3GPP2 are addressed, the LTE access inside the EPS were not yet mature enough when the standard was finalized.

This study [12], performed for the MEDIEVAL project, consisted in defining an appropriate mapping between the abstract interface provided by the MIH_LINK_SAP primitives and the related procedures available in the EPS and LTE protocols. It corresponded to the function performed by the Link Interface component described in Section 4.3.2, associated to the managed LTE wireless access. Moreover, this study was extended to include EURECOM's LTE platform and its direct interconnection between the Access Stratum and the IP protocol stack, implemented as a middleware above the RRC protocol [92]. The result has been submitted to the IEEE 802.21 [18] and approved to be included in a next revision of the standard.

In the EPS, the control part of the system is simpler and more efficient than in UMTS. Figure 64 shows the control plane protocol stacks linking the MT (or User Equipment, UE), to its eNodeB, acting as the PoA, and the associated MME, acting as an AR or Point of Service (PoS). This control plane provides procedures for the network access connection, disconnection, address activation, mobility and user plane resource allocation.

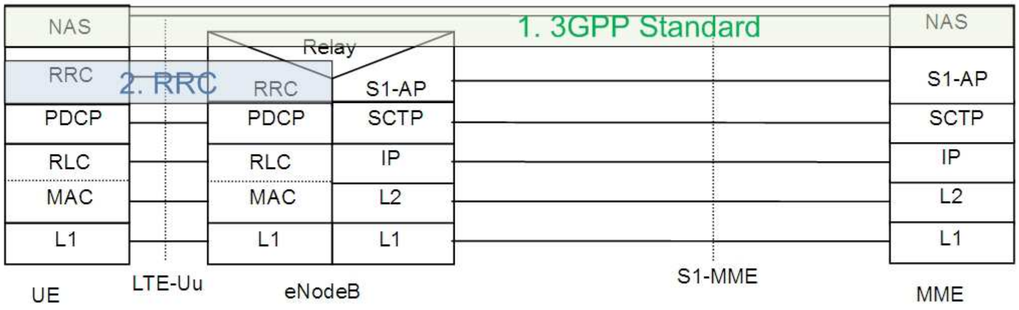


Figure 64: EPS control plane for access network interfaces, from [34]

Since the standard 3GPP protocols are not easily accessible, the 802.21 primitives must be mapped at Layer 3 level, e.g., to the Non Access Stratum (NAS) protocol. This is what is first proposed in Figure 64 (green box). However, the EURECOM experimental platform allows interacting directly with the RRC protocol operating at Layer 2, so it seems interesting to bypass the 3GPP Layer 3 and perform the mapping directly at this lower level. This is what is proposed in a second step in the figure (blue box).

Figure 65 illustrates how the MIH reference model has been applied to the EPS system.

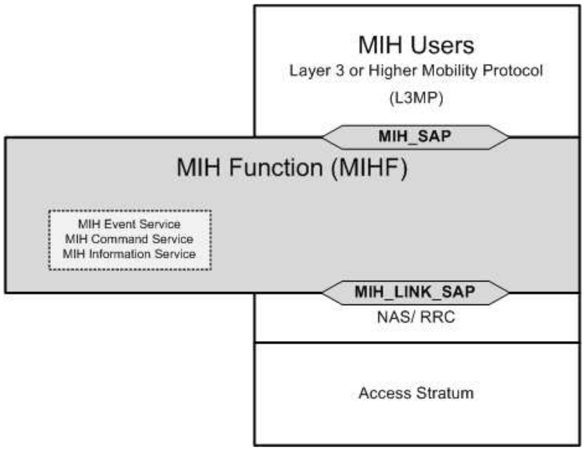


Figure 65: MIH reference model for EPS systems

<u>Mapping with the NAS and RRC protocols</u>

The table below shows the proposed mapping and is followed by its rationales. A very short summary of the 3GPP procedures and primitives used in this table is provided in ANNEX B.

Primitives	NAS protocol	LTE/RRC procedure
Link_Detected	Attach MT Signal Quality*	System Information
Link_Up	Attach Activate Default EPS bearer context Modify EPS bearer context	RRC Connection establishment RRC Connection re-establishment RRC Connection reconfiguration
Link_Down	Detach Deactivate EPS bearer context ESM Status	RRC Connection reconfiguration RRC Connection release
Link_Parameters_Report	Tracking Area Update	Measurement report
Link_Going_Down	N/A	N/A
Link_Handover_Imminent	N/A	N/A
Link_Handover_Complete	Tracking Area Update Activate Default EPS bearer context Activate Dedicated EPS bearer context	RRC Connection reconfiguration

Link_PDU_Transmit_Status	N/A	N/A
Link_Capability_Discover	N/A	N/A
Link_Event_Subscribe	Packet Domain Event reporting*	Measurement configuration
Link_Event_Unsubscribe	Packet Domain Event reporting*	Measurement configuration
Link_Get_Parameters	EPS QoS Dynamic parameters* MT Signal Quality*	Measurement configuration
Link_Configure_Thresholds	Modify EPS bearer context	Measurement configuration
Link_Action / Disconnect	Detach Deactivate EPS bearer context	RRC Connection release
Link_Action / Low Power	MT Set Functionality*	N/A
Link_Action / Power Down	Detach	RRC Connection release
Link_Action / Power Up	Attach	RRC Connection establishment

Table 15: Mapping from MIH to NAS and RRC protocols

Rationales for the mapping with the NAS protocol

This mapping is based on 3GPP service procedures and commands.

When the 802.21 primitive involves some interaction between the mobile terminal and the network, this mapping refers to NAS procedures. The NAS protocol [93] has been selected because it is the EPS equivalent to the Layer 3 protocols that were used for the MIH_3GLINK_SAP mapping in the current version of the standard. The NAS procedures are used by the mobility and session management functions between the terminal and the MME in the EPS. When relevant, an equivalent signalling is defined between the eNodeB or PoA and the MME, as part of the S1-AP protocol [94]. The first two columns of the mapping table (Table 15) list the NAS procedures rather than primitives, in order to remain generic and compatible with both sides: MT and network nodes. From a general point of view, the end of a NAS procedure may trigger an MIH event, while NAS procedures should be triggered by the reception of MIH commands.

When the 802.21 primitive implies a local action in the mobile only, the corresponding local AT command defined for the operations inside the MT, as specified in [95], is used (AT means ATtention; this two character abbreviation is always used to start a command to the modem in the terminal). These commands are marked with a (*) in the mapping table. This mapping would have to be extrapolated for network nodes, since equivalent commands are usually vendor-implementation dependant in the network equipments, and thus not standardized.

In the same manner as in the 3GPP mapping proposed by the 802.21 specification, a NAS procedure or AT command can be mapped to more than one MIH primitive.

Rationales for the mapping with the RRC protocol

When the RRC sub-layer [92] is reachable (e.g., in the EURECOM platform), it is also interesting to establish a direct mapping between the 802.21 primitives and the corresponding procedures of the RRC protocol for system broadcast, connection setup or radio resource configuration. This correspondence is given as the third column of Table 15.

In summary, the application case presented in this section has shown an example of Link Interface specification for the integration of the LTE wireless access to the IEEE 802.21 standard, in a manner similar to what it would be to the MIS in the CCF system.

6.6 Summary

Five partial applications of the framework to real systems and projects have been explained in this chapter. The abstraction from the technology specificities has been applied to the context of seamless mobility and QoS management in heterogeneous networks, to cross-layer geo-casting and to enhance the support of video services in future mobile operator networks. This abstraction has also been upgraded to include in a detailed manner the recent LTE wireless access in the MIH standard. The principle governing the Generic Service Enablers has been exploited to control the GeoNetworking dissemination of messages in the ITS environment or in the context of public warning systems, and to design the management layer of the ITSS, including its communication profile selection service. Finally, the collection and distribution of system variables across the different layers under the control of a dedicated entity has been used for the Management layer of the ITSS.

The diversity of the possible application situations for each of the main CCF principles shows the generality of the approach and proves that they can be beneficial by themselves individually. However, in most cases, at least two of them have been combined and coordinated, clearly improving the overall efficiency of the global system. The CCF, which combines all three principles, is thus able to neatly solve the problem raised by the break of the applications sessions when moving across independent networks and to allow a user-centric, but still network-assisted, approach for the attachment to the most convenient access network.

CHAPTER 7 - CONCLUSION AND PERSPECTIVE

7.1 Conclusion and Benefits of the Framework

The evolution of mobile communications has generated new challenges for the design of the connectivity functions of future terminals. One of them is the possibility given to the user to roam across heterogeneous and non-federated access networks without having to restart manually his applications. Another one is the capability to choose the best available network at a certain time, taking into account not only the received signal strength, but also other parameters such as user preference, its environment variables or the requirements of the running application.

To assess the validity of these issues, some experiments have been performed on a set of applications commonly used in a real terminal and were reported at the beginning of this study. Next, several candidate technologies were analysed to identify what they could bring to the target solution and what were their limitations. The heterogeneous networks and IEEE 802.21 MIH services, together with several available schemes for mobility, have been studied. They were completed by surveys of techniques for the access network selection, for the handling of multimode terminals and for the architectural blueprint of autonomous systems. The objective was to map their main concepts towards a new architecture that could satisfy our requirements. The vertical application domain of Intelligent Transport Systems, which constitutes a notebook case for these issues, has been presented. Based on this knowledge, three typical scenarios were determined, which allowed to highlight the mobile session continuity, the usage in vehicular communications and in emergency public warning notifications. They were used to build a set of requirements that would lead the design of the target system. The main requirement was to enable, independently from the network provider, the roaming of a Mobile Terminal across heterogeneous networks, with a new architecture affecting the terminal only. The mobile must however be able to comply with the existing network policies and maintain the operation of existing applications and network protocols. It must provide self-configuration while authorizing the device user to change it in a simple manner. Moreover, the process must remain light in terms of battery or processing power consumption. This part of the study has been complemented by the reproduction of the applications behaviour in a simulated environment.

As a result, a cross-layer and integrated framework, the Connectivity Control Framework, has been designed following the layered architecture of autonomous systems. Its layout is based on three main principles: *(i)* an abstraction layer which hides the network specificities to the rest of the framework and includes in addition the support of other hardware devices such as positioning systems or diverse sensors; *(ii*) some coordinated generic service enablers responsible of specific tasks and taking care of the various functions necessary to handle the terminal connectivity; *(iii)* a cross-layer agent which maintains and shares the knowledge acquired by the other components of the framework. Several small scenes, derived from the main scenarios, are detailed to demonstrate the operation of the framework, followed by a definition of the system interfaces and of its internal components. To assess its benefits, a simulation model has been developed, implementing the whole framework and experimenting its behaviours in the testing environment previously setup.

The results show that the requirements that had been drawn initially are fulfilled. The CCF is restricted to the mobile terminal and has no impact on the mobile network infrastructure, while maintaining full compatibility with existing networking standards. Under some conditions, it only impacts the Correspondent Node, which has to implement a subset of the generic service enabler which handles the application session.

The extension of the abstract mechanisms of the IEEE 802.21 standard allows the upper layers to remain unaware of the technology, for any type of control and not only for handovers. The mobile attachment and behaviour can be ruled in an easier and more accurate manner. The access network selection is based on a large set of parameters, reflecting the user's context. Combining the generic service enablers with the cross layer interactions enhances the flexibility of the whole framework. Moreover, it provides a clear uncoupling between the applications and the access network and any existing application can benefit from this concept. Thanks to the socket mapping mechanism implemented in the Session GSE, the application or the user does not have to recover or restart after a network change. Connectivity disruption is short enough to be resolved by a short buffering of the data packets during the handover. The packet delay is not degraded because this concept implies only a small additional processing directly at socket level. Even though additional functionalities have been installed in the mobile terminal, the additional power consumption is limited. Most of the time, the monitoring task is the only task active and it operates as a background task. Moreover, it already exists in the current devices. It wakes up the rest of the framework on the control plane only when an abnormal condition is encountered, in which case an event is triggered. This usually happens in a limited number of occurrences. Putting the periodic polling and monitoring functions at its edge, provides a reduced need in processing power and in battery consumption. The autonomous behaviour and the integrated system coordination reduce the switching time between two known networks and the interaction with the mobile user, increasing his comfort and QoE. The communication protocols are not impacted, the interactions with the network are not affected; the framework is compatible with existing data security and integrity mechanisms implemented at lower layers, which means that this concept has a minimum dependency on the technologies used by the various operators, including the usage of IPv4 or IPv6 protocols.

Part of the framework has been applied to a wide range of application cases on real systems. The abstraction principle has been used to enable seamless mobility experiments using a LTE-like cellular access for video application delivery. It has been applied to define the internal functions of the Management layer in an ITSS or for cross-layer geo-broadcasting in the ITS and emergency notification use case. This diverse set of application cases demonstrates the generality and benefit of the chosen approach, while opening the path to a fast acceptability. The cost of its implementation can be shared by the mobile manufacturers as part of an operating system upgrade and passed on to the millions of terminals sold each year, while providing a higher satisfaction of the mobile users from a qualitative perspective.

Moreover, 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.

7.2 Perspectives and Future Work

This study has investigated comprehensively the distribution of functions and the integration and orchestration of the system proposed. For each individual component, existing solutions that satisfied the main requirements have been selected and necessary enhancements described. Due to the wide range of the system defined, several topics could not be analysed in details which may potentially be improved and would require a follow-up of this study. This applies to a more precise analysis and definition of the generic service enablers introduced in the CCF: the network access selection, the session management and the Network layer support along the on-going activities at the IETF (MIF) or the OMA standardization groups.

Another topic would be to extend the objectives of the CCF to the green operation of the terminal, capitalizing on the support of hardware devices by the MISF. A short snapshot of what this study could bring has been mentioned in Section 4.3.2, with the coordination of the battery, solar cells and external power supply according to the internal context in the MT.

An extension could also involve the cooperation with mobile network entities, going beyond the handover and other mechanisms such as the MIIS or the ANDSF, and fully exploiting the capabilities brought by the usage of the recent cloud networking.

Finally, as mentioned at the end of Chapter 5, the simulation model could be enhanced with functions that have been streamlined in this study to demonstrate the full extent of the framework.

ANNEX A - DETAILED INTERFACES DEFINITION

This Annex provides the detailed description and parameters for the primitives defined in Section 4.2.3.2.

1. User Interaction ←→ Connectivity Agent – INT1

- Get_Network_Selection_Validation.request/confirm
  - Function : validate network selection decision made by NAGSE
  - Parameters : ordered list of selected networks (request), Selected network and L2 credentials if relevant (confirm)
- Get_Credentials.request/confirm
  - Function : obtain credentials to register and authenticate in the network.
  - Parameters : L2/NETWORK, Net Id (request), credentials (confirm)
- Get_Preferences.request/confirm
  - Function : Configure or modify pre-defined policies and precedence order
  - Parameters : Type: Network/Technology (request), Network policies / technology precedence, condition (confirm)

2. Cognitive Manager ←→ Generic Service Enablers – INT2

NAGSE

- Get_Available_Networks.request/confirm
  - Function : Obtain ordered list of available networks
  - Parameters : N/A (request), ordered list of networks (confirm)
- Get_Application_Connectivity_Validation.request/confirm
  - Function : Check application requirements vs. existing connections
  - Parameters : App Id (request), App Id , Validity answer (confirm)

MGSE

- Action_Connect_Network.request/confirm
  - Function : Connect the MT to network X
  - Parameters : Net Id (request), connection status (confirm)
- Event_System.indication
  - Function : Signal a system event (from MISF interface)
  - Parameters : Event type, event attributes

SGSE

- Action_Authentication_Check.request/confirm
  - Function : Validate Net authentication
  - Parameters : Net Id (request), authentication status (confirm)
- Action_Application_Session_Start.request/confirm
  - Function : Start session for application X on network Y
  - Parameters : App Id, Net Id (request), status (confirm)

- Action_Application_Session_Transfer.request/confirm
    - Function : Transfer session for application X on network Y; switch the data traffic to the new socket opened.
    - Parameters : App Id, Net Id (request), status (confirm)
- Action_DTN_Buffering_Start.request/confirm
    - Function : Start DTN buffering for an application which is about to be transferred
    - Parameters : App Id (request), buffering activation status (confirm)

3. Generic Service Enablers ←→ MIS Function – INT3 – INT4 – EXT5

Following the MIH model, all primitives are forwarded to the Link Interface (INT4) and mapped to device procedures (EXT5)

NAGSE

- Link_Action.request/confirm [Monitor]
    - Function : monitor available networks from network interfaces and report capabilities and status back to the NAGSE. Reporting content includes networks and Access Points available
    - Parameters : All/specific technology (request); list of networks detected with link and network information, including Net Id, technology, L2 authentication required, signal quality, capabilities (confirm)
    - Wireless Link mapping : network scan (WLAN), Listen to BCH (LTE)

MGSE

- Link_Action.request/confirm [Attach / Detach]
    - Function : Attach to network X; Power-up interface or switch to the defined AP (similar primitive for detach)
    - Parameters : Net Id, Link Id, APN (request), status, signalling path established (confirm)
    - Wireless Link mapping : activate connection (WLAN), wake-up connection (LTE)
- Link_Action.request/confirm [Authenticate]
    - Function : Execute L2 authentication
    - Parameters : Net Id, credentials from the LIB are inserted by the MISF (request), status, signalling path established (confirm)
    - Wireless Link mapping : activate connection (WLAN only)
- Link_Action.request/confirm [Parameters]
    - Function : Retrieve a set of parameters from the network or a device (location, address, remaining power...)
    - Parameters : Net Id, list of parameters types (request), Net Id, list of {parameters types, parameters values} (confirm)
    - Wireless Link mapping : Get Parameter command
- Link_Configure.request/confirm [Link_measurements]
    - Function : Configure periodic or event-based measurements on the link
    - Parameters : Net Id (request), status (confirm)
    - Wireless Link mapping : Start periodic link measurement / subscribe to the requested event
- Link_Report.indication [System event]

    - Function : Signal a network or device event in the terminal
    - Parameters : Net Id / Device Id, Event type, optionally additional attributes
    - Wireless Link mapping : New access detected/ signal quality decreasing rapidly / network lost ...
- Link_Report.indication [Link Parameters]
    - Function : Report periodic link measurements previously configured
    - Parameters : Net Id, link measurements
    - Wireless Link mapping : Measurement report

SGSE

- Link_Action.request/confirm [Activate Resources / Deactivate Resources]
    - Function : Allocate QoS resources in the network; request resource allocation with the QoS associated to the application
    - Parameters : Session Id, Net Id, QoS requested (request), status, QoS parameters (confirm)
    - Wireless Link mapping : N/A (WLAN), Define QoS for data bearer (LTE)
- Link_Configure.request/confirm [Session Measurements]
    - Function : Configure periodic or event-based measurements for the session (data traffic)
    - Parameters : Net Id, Session Id (request), status (confirm)
    - Wireless Link mapping : Start periodic traffic quality measurements / subscribe to the requested event
- Link_Report.indication [Session Parameters]
    - Function : Report periodic session traffic measurements previously configured.
    - Parameters : Net Id, Session Id, traffic measurements
    - Wireless Link mapping : Measurement report

4. Connectivity Agent / MISF ←→ Cross-Layer Agent – INT5 – INT6

- Set network preferences
    - Function : Store or update user network preferences;
    - Parameters : Net Id, approved/Rejected, L2 authentication credentials
- Set technology preferences
    - Function : Store or update user technology preferences;
    - Parameters : technology type, precedence, condition
- Get application requirements
    - Function : retrieve stored knowledge on an application;
    - Parameters : App Id, TFT (QoS, bit rate), multicast enabled, preferred technology, priority
- Set algorithm parameters
    - Function : Store decision algorithm parameters;
    - Parameters : input values for the algorithm, result of algorithm
- Set link parameters
    - Function : Store or update the parameters of a wireless link;
    - Parameters : Link Id, link type, signal quality, Net Id, L2 authentication, signal quality, other link level parameters
- Set connection parameters
    - Function : Store or update the parameters of a connected access network;

    - Parameters : Net Id, link Id, capabilities, connection status, envA, other connection level parameters
- Set session parameters
    - Function : Store or update the parameters of a specific session;
    - Parameters : App Id, session Id, connected Net Id, PA, envA, destination address, port numbers, QoS, status of DTN mechanism, other session level parameters
- Get link information
    - Function : retrieve stored knowledge on a specific wireless link;
    - Parameters : Link Id / link type, signal quality, other link level parameters
- Get connection information
    - Function : retrieve stored knowledge on found networks;
    - Parameters : Net Id / (unknown, known/approved, known/rejected), network attachment policies, related user preferences, other connection level parameters
- Get session information
    - Function : retrieve stored knowledge on a specific application session;
    - Parameters : App Id, session Id, PA, envA, transport protocol, other session level parameters
- Set periodic measurement
    - Function : store a measurement report and update the related fields in the LIB;
    - Parameters : parameters measured

ANNEX B - SUMMARY OF 3GPP PROCEDURES USED FOR THE MAPPING WITH 802.21

This Annex provides the summary of the 3GPP procedures associated with the MIH mapping presented in Section 6.5.

Summary of the NAS procedures used (from [93])

Attach: The attach procedure is used to attach to an Evolved Packet Core (EPC) Network for packet services in EPS.

Detach: The detach procedure is used by the UE to detach from EPS services, by the network to inform the UE that it is detached or by the network to inform the UE to re-attach to the network and re-establish all connections.

Activate Default EPS bearer context: The purpose of the default bearer context activation procedure is to establish a default EPS bearer context between the UE and the EPC. The default bearer context activation procedure can be part of the attach procedure. The default EPS bearer context does not have any TFT assigned during the activation procedure. This corresponds to using a match-all packet filter. The network may at any time after the establishment of this bearer assign a TFT to the default EPS bearer and may subsequently modify the TFT or the packet filters of this default bearer.

Activate Dedicated EPS bearer context: The purpose of the dedicated EPS bearer context activation procedure is to establish an EPS bearer context with specific QoS and TFT between the UE and the EPC.

Modify EPS bearer context: The purpose of the EPS bearer context modification procedure is to modify an EPS bearer context with a specific QoS and TFT.

Deactivate EPS bearer context: The purpose of the EPS bearer context deactivation procedure is to deactivate an EPS bearer context or disconnect from a Packet Data Network (PDN) by deactivating all EPS bearer contexts to the PDN.

ESM Status: The purpose of the sending of the EMM STATUS message is to report at any time certain error conditions detected upon receipt of EMM protocol data. The EMM STATUS message can be sent by both the MME and the UE.

Tracking Area Update: The MME knows the location of the UE with the granularity of a few cells, called the Tracking Area (TA). The TA update procedure is always initiated by the UE and is used for purposes such as updating the registration of the actual TA of a UE in the network, updating certain UE specific parameters in the network, recovering from certain error cases.

Summary of the AT commands used (from [95])

Packet Domain Event reporting: This command enables or disables sending of unsolicited result codes, such as network detach, context activation, modification or deactivation.

EPS QoS Dynamic parameters: The execution of this command returns the QoS parameters (QCI, DL_GBR, UL_GBR, DL_MBR, UL_MBR) of an established PDP Context.

MT Signal Quality: The execution of this command returns the received signal quality parameters: RSSI, BER (channel bit error rate), RSCP, Ec/No (energy per chip divided by the noise power), RSRQ, RSRP.

MT Set Functionality: This command allows to select the level of functionality in the MT. Level "full functionality" is where the highest level of power is drawn. "Minimum functionality" is where minimum power is drawn.

Summary of the RRC procedures used (from [92])

System Information: Broadcast at the cell level of system information, including NAS common information, cell parameters, neighbouring cell information or common channel configuration, ...

RRC Connection establishment: The purpose of this procedure is to set-up the connection of the radio interface. This procedure is also used to transfer the initial NAS dedicated information/ message from the UE to the eNodeB.

RRC Connection re-establishment: The purpose of this procedure is to re-establish the RRC connection when a valid UE context is known in the network.

RRC Connection reconfiguration: The purpose of this procedure is to modify an RRC connection, e.g. to establish/ modify/ release radio channels, to perform handover, to setup/ modify/ release measurements.

RRC Connection Release: The purpose of this procedure is to release the RRC connection, which includes the release of the established radio bearers as well as all radio resources.

Measurement configuration: In the RRC procedures, measurements performed by the UE are reported to the network. The UE reports the measurement information in accordance with the measurement configuration as provided by the eNB. The eNB provides this configuration, applicable for a connected UE, using the RRCConnectionReconfiguration message. In the platform, some measurements can also be configured for local reporting to the upper layers of the UE.

Measurement report: The purpose of this RRC procedure is to transfer measurement results from the UE to E-UTRAN. In the platform, the reporting can also be performed locally towards the upper layers of the UE.

RÉSUMÉ ÉTENDU

1. Motivation

L'évolution des communications mobiles a généré de nouveaux défis pour la conception des fonctions de connectivité des futurs terminaux. La tendance dans les technologies mobiles a consisté en la conception d'appareils multimodes, avec un nombre croissant d'interfaces, et capables de se connecter à n'importe quel réseau disponible. Un terminal multimode mobile (MT, Mobile Terminal) est un appareil informatique, de type ordinateur portable, smartphone, tablette ou ordinateur de bord, équipé de plusieurs interfaces réseau et capable d'établir des communications à travers une ou plusieurs de ces interfaces à un moment donné. Du fait de l'apparition de différentes normes pour les technologies sans fil, amenant des propriétés différentes, les réseaux mobiles sont devenus hétérogènes, intégrant plusieurs types de technologies d'accès dans le même domaine administratif. Ces accès offrent des caractéristiques de connectivité différentes pour les protocoles et les applications utilisateur et demandent une adaptabilité et un contrôle supplémentaires dans les niveaux supérieurs de la pile de protocole, afin de permettre une itinérance transparente à travers les différents accès. L'itinérance et la mobilité dans les réseaux hétérogènes font donc partie des activités cruciales de l'étude des communications mobiles. Actuellement, en l'absence de fédération entre deux opérateurs de téléphonie mobile et de déploiement d'un mécanisme de mobilité spécifique dans le réseau sans fil, l'itinérance signifie très souvent que la session servant l'application en cours d'exécution est cassée et doit être redémarrée manuellement, au prix d'une perte de données, sauf si l'application est conçue pour se récupérer par elle-même.

La plupart des applications existantes les plus utilisées sont basées sur le protocole TCP (Transmission Control Protocol) pour leurs transferts de données. TCP a été réalisé sous la forme d'un protocole statique, donc lorsque l'identifiant de l'une de ses extrémités, à savoir son adresse IP (Internet Protocol) et son port de socket, change, la connexion TCP est coupée. Certaines applications se bloquent ou stoppent leur exécution tandis que d'autres sont capables d'établir une nouvelle connexion TCP et de reprendre leur activité. Finalement, tout dépend de la façon dont l'application a été développée. Un autre défi à résoudre est le choix du réseau d'accès optimal par le terminal multimode. Actuellement, un smartphone privilégie une connexion Wifi connue lorsqu'elle est détectée, transférant systématiquement tout le trafic de données sur cet accès. En l'absence de Wifi, il transfère tout ce trafic via le réseau cellulaire s'il est configuré. Cela peut conduire à des comportements bizarres ou indésirables. De plus, de nouvelles technologies d'accès, pour les communications inter-véhiculaires par exemple, sont sur le point d'être déployées, créant un niveau supplémentaire de complexité pour la sélection dynamique du réseau d'accès optimal dans un contexte donné. La solution actuelle, binomiale et statique, est donc appelée à devenir rapidement trop simpliste et limitée par rapport à la complexité de l'environnement mobile et aux contraintes de connectivité prévisibles dans un proche avenir.

L'objectif principal de cette thèse est de s'attaquer au problème de la défaillance de l'application lors du changement de réseau d'accès sans support de mobilité, en particulier lorsque le protocole de transport TCP est utilisé. Le résultat permettra à une personne utilisant un appareil mobile de se déplacer de manière transparente entre des environnements sans fil hétérogènes et non fédérés. Un tel environnement peut être un réseau mobile d'opérateur, le hotspot du café local ou le réseau privé de l'utilisateur. Suivant la considération que les nouvelles techniques et algorithmes qui modifient les entités existantes dans le réseau sont difficiles à déployer, cette étude a adopté une approche novatrice pour résoudre le problème, en proposant d'appliquer toutes les modifications nécessaires au terminal mobile uniquement et en laissant le réseau totalement inchangé.

Plusieurs techniques de mise en réseau tels que les services MIH (Media Independent Handover), la gestion de la mobilité, la sélection d'accès au réseau ou les systèmes autonomes peuvent contribuer à atteindre cet objectif. Cependant, aucune de ces techniques n'est en mesure de fournir une solution complète par elle-même et de prendre en charge efficacement les divers périphériques situés dans le terminal. Notre objectif est donc de les intégrer dans une structure unique qui fournira aux futurs systèmes un contrôle souple et optimisé de leur fonctionnement multimode.

2. Technologies de Référence et Défis

Certaines technologies existantes abordent partiellement la sélection d'accès hétérogène et dynamique. Elles seront présentées dans ce chapitre. Les services MIH ont été mis en place pour faciliter les handover de terminaux. Des mécanismes de mobilité ont été introduits pour résoudre les interruptions de connectivité à tous les niveaux de la pile de protocole. De nombreuses contributions techniques présentent ou passent en revue des études de sélection d'accès, le plus souvent du point de vue du terminal. La conception des systèmes autonomes peut être utilisée comme source supplémentaire d'inspiration pour améliorer la résilience et le dynamisme de ces systèmes. Ce chapitre se termine par une présentation des défis spécifiques rencontrés par les communications inter-véhiculaires.

Le fonctionnement des appareils multimodes dans les réseaux hétérogènes peut devenir très complexe si chaque technologie d'accès doit être adressée directement et séparément par les entités de mise en réseau.

La norme IEEE 802.21 propose trois Services Indépendants du Media différents [23], qui offrent aux protocoles de gestion des couches supérieures des déclencheurs génériques, l'acquisition d'informations et les outils nécessaires pour effectuer les handovers. Le service d'événements (MIES) fournit le cadre nécessaire pour gérer les événements du réseau et pour rendre compte de l'état dynamique des différents liens. La fonction de commande (MICS) permet de contrôler le comportement des liens alors que le service d'information (MIIS) distribue les informations et les règles liées à la topologie à partir d'une bibliothèque située dans le réseau. Une architecture inter-couches est définie où la fonction MIH (MIHF), illustrée sur la Figure 1, sert de relais entre *(i)* les entités spécifiques aux médias de la couche Liaison reliées par le MIH_LINK_SAP (Service Access Point) et *(ii)* les entités agnostiques des médias de la couche supérieure, ou utilisateurs MIH, connectées par le MIH_SAP.

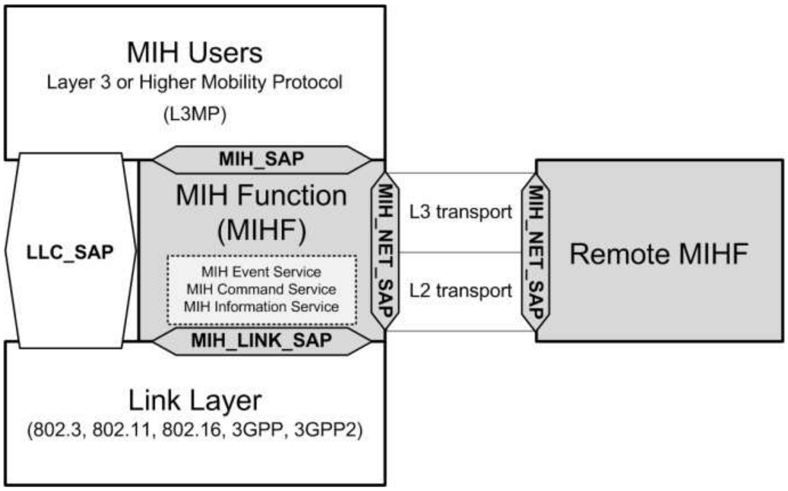


Figure 1: Modèle de Référence pour le Media Independent Handover

Actuellement, la norme IEEE 802.21 prévoit des mécanismes permettant de contrôler les interfaces réseau d'un terminal multimode de manière abstraite et indépendante du média. Cependant, elle comporte plusieurs limitations importantes. Elle ne permet actuellement que les handovers et concerne exclusivement les interfaces de mise en réseau. Elle offre donc la possibilité d'être étendue pour supporter un ensemble plus important de services et de périphériques. Cette extension sera un axe principal pour la conception de la solution cible.

Lorsqu'un appareil se déplace hors de sa zone de routage d'origine, il ne peut pas conserver son adresse IP et l'exécution des sessions de données est interrompue. Les paquets sont encore transmis le long de l’ancienne route mais ne sont plus en mesure d'atteindre le mobile. Pour résoudre ce problème, les groupes de l’IETF abordent la question de la mobilité à différents niveaux de la pile de protocole. La couverture des solutions est très vaste aussi: au niveau du périphérique, de la couche Transport, de la session, de l'application, ou encore plus récemment, du flux. L'objectif est de concevoir des protocoles capables de survivre aux interruptions de connectivité du terminal. Des mécanismes et protocoles parmi lesquels se trouvent Mobile IP, Mobile IP rapide, PMIP (Proxy Mobile IP), mSCTP (mobile Stream Control Transmission Protocol) ou SIP (Session Initiation Protocol) ont été proposés [28]. Ces solutions, soit ne résolvent pas le problème du multi-domaine, soit exigent des changements importants de l'architecture du réseau et de ce fait, se heurtent à une forte réticence pour leur déploiement. De leur côté, les systèmes cellulaires gèrent la mobilité avec des protocoles et des procédures propriétaires du 3GPP (3rd Generation Partnership Project) [34], parfois adaptés à partir des protocoles de l'IETF.

A côté de ces mécanismes, une technique intéressante est basée sur des réseaux tolérants aux délais (DTN, Delay Tolerant Networks) qui permettent aux mobiles de survivre à de longues interruptions de connectivité. L’impact de l’intermittente de la connectivité est résolu en utilisant une commutation de message de type store-and-forward (stockage - retransmission) [39] [40]. Un message entier, ou découpé en morceaux appelés "bundles", est transféré entre des nœuds de stockage permanent, ou nœuds DTN, et maintenu en mémoire jusqu'à ce qu'ils soient en mesure de transmettre le bundle au nœud DTN suivant. Cette fonctionnalité est fournie par une "couche bundle", insérée entre l'application et la couche de transport, comme le montre la Figure 2. Le protocole TCP est utilisé comme protocole de transport uniquement pour assurer la fiabilité de la communication sur un segment de réseau donné. Les nœuds DTN assurent la terminaison des protocoles de transport à la couche Bundle, ce qui rend cette architecture tolérante aux délais et aux problèmes de connectivité.

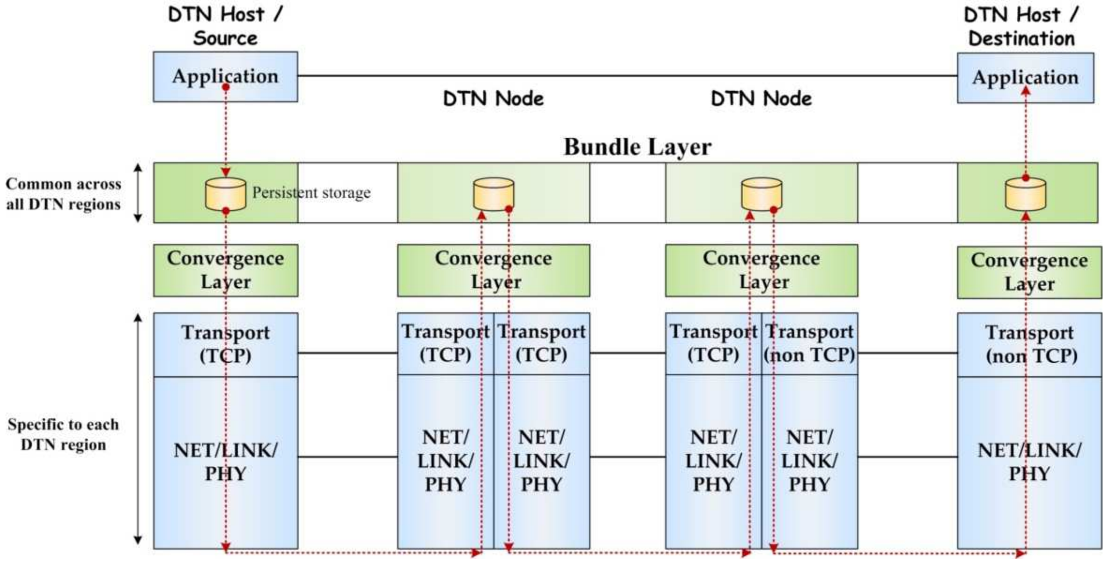


Figure 2: Commutation de message store-and-forward dans les nœuds DTN (cf. [39])

De nombreux mécanismes ont été conçus qui traitent de la mobilité à l'intérieur d'un domaine unique ou entre deux domaines apparentés. Ils introduisent tous de nouvelles entités fonctionnelles ou de nouveaux protocoles dans le réseau. Certains nécessitent des tunnels qui réduisent l'efficacité du trafic et donc la bande passante laissée à l'application elle-même. De plus, ils s'appuient souvent sur des entités spécifiques détenues par l'opérateur du réseau et ne fonctionnent pas correctement lors du déplacement vers ou à partir d'un réseau n'est pas reconnu par cet opérateur. Finalement, la continuité de la session dépend de la capacité de l'application à récupérer par elle-même des interruptions.

Un défi majeur pour la gestion de la connectivité dans les réseaux hétérogènes avec des terminaux multimodes consiste à obtenir une sélection de réseau d'accès optimale.

La Figure 3 montre un terminal mobile situé à la limite de deux réseaux d'accès LTE et Wi-Fi, qui doit décider quel est le chemin le plus efficace pour atteindre le serveur d'applications situés derrière l'Internet. Cette procédure de décision, dont l'objectif est de sélectionner un nouvel accès sans fil ou d'en rajouter un à ceux déjà utilisés, peut être divisée en quatre étapes: *(i)* la collecte de données d'entrée, *(ii)* l'algorithme de sélection et de prise de décision, *(iii)* la validation des paramètres de sortie, et éventuellement, *(iv)* la confirmation de l'utilisateur.

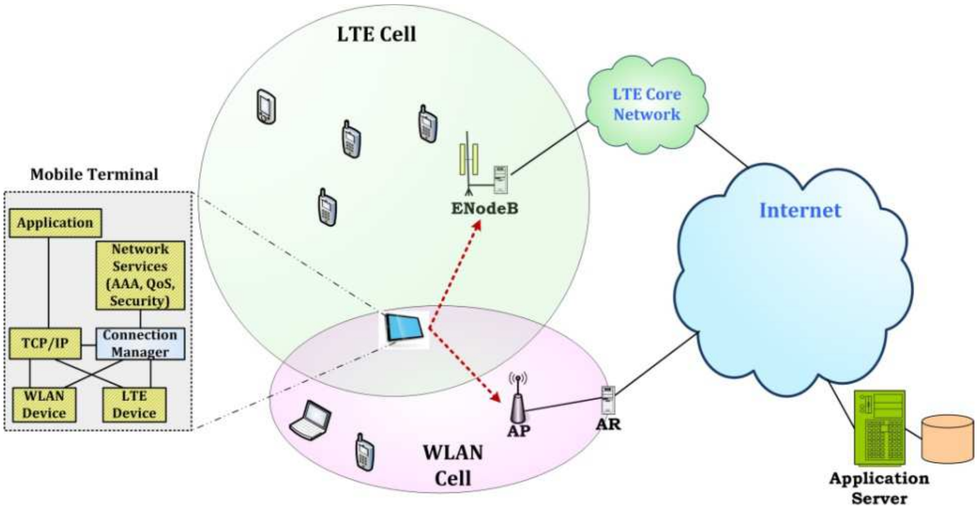


Figure 3: Exemple de configuration du réseau impliquant la sélection d'accès au réseau

La tendance dans la littérature est de prendre cette décision dans le terminal, grâce à une nouvelle entité appelée le gestionnaire de connexions (Connection Manager ou CMGR) [42] [43]. La décision du CMGR est basée sur le contexte propre au MT (exigences des applications, tarifs d'abonnement, certificats de sécurité, qualité du signal...). Elle est parfois améliorée grâce à des paramètres ou des règles récupérés dans le réseau. Lorsque toutes les entrées ont été réunies, un algorithme de sélection est exécuté. Les auteurs de [59] classent les différentes stratégies de décisions existantes, y compris les stratégies centrées sur l'utilisateur, en prenant en compte les préférences des utilisateurs en termes de coût et de qualité de service ou les stratégies capable de résoudre des problèmes de prise de décision multi-attributs (Multi-Attribute Decision Making, MADM). L’approche se base sur des méthodes bien connues telles que le SAW (Simple Additive Weighting, pondération additive simple), le TOPSIS (Technique for Order Preference by Similarity to Ideal Solution, Technique de tri préférentiel par similarité à la solution idéale) ou le WP (Weighted Products, produits pondérés). La plupart des algorithmes proposés nécessitent une exécution continue et consomment donc une partie importante de la puissance de traitement du système. De plus, la décision n'est jamais confrontée à l'opinion de l’utilisateur final, en particulier lorsque des informations manquantes ont pu mener à un mauvais choix. Quand cette décision est prise avec le point de vue du réseau, l'objectif est de fournir un accès évolué à des terminaux multimodes mobiles dans un réseau large et hétérogène, tout en ménageant la charge du cœur de réseau. Dans ce cas, les préférences de l'utilisateur ne sont pas prises en considération, de même que les accès qui n’appartiennent pas à l'opérateur.

S’occupant du fonctionnement du MT d’un point de vue global, l'Open Mobile Alliance (OMA) a récemment publié des normes proposant une API banalisée pour les interfaces synchrones et asynchrones entre le CMGR et les applications et interfaces utilisateurs dans le terminal [52]. Cette norme introduit des composants fonctionnels qui fournissent des services génériques aux applications propriétaires dans le mobile. Cependant, les interfaces basses avec les différents périphériques matériels restent dépendantes de leurs caractéristiques de réalisation. De plus, ces services et fonctions ne sont pas inter-corrélés et sont exécutés comme des entités distinctes.

D'autres études de conception des architectures futures introduisent un plan cognitif totalement nouveau, où l'environnement est détecté et observé, conduisant à l'acquisition d’une connaissance qui est exploitée dans une capacité nouvelle d'autogestion [64]. Ces systèmes autonomes (Autonomous Systems, AS) sont adaptables pour faire face à des situations imprévues ou à des changements dynamiques se produisant dans leur environnement. L'autogestion est effectuée principalement selon certaines règles internes et sans nécessiter l’implication d'un utilisateur humain. Le système fonctionne en exécutant des boucles de contrôle intelligentes [62]. Il détecte son environnement de fonctionnement, travaille avec des modèles qui analysent son propre comportement dans cet environnement, et, sur la base des règles existantes et des connaissances apprises, déduit les mesures appropriées pour adapter son état ou son comportement.

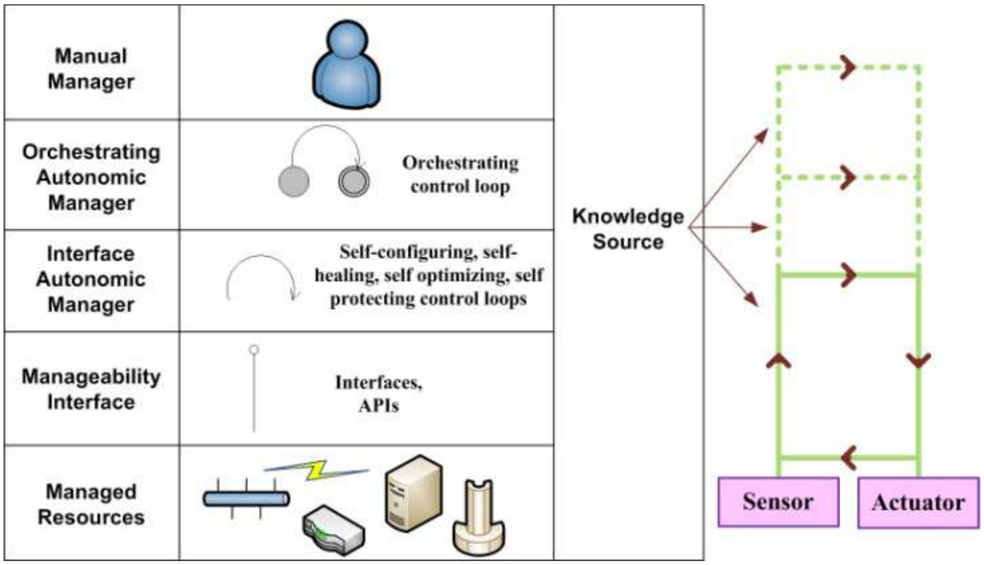

Manual Manager
Orchestrating Autonomic Manager
Orchestrating control loop
Interface Autonomic Manager
Self-configuring, self-healing, self optimizing, self protecting control loops
Manageability Interface
Interfaces, APIs
Managed Resources
Knowledge Source
Sensor
Actuator

Figure 4: Hiérarchie de contrôle autonome

L'architecture AS est structurée selon une hiérarchie de décision, présentée sur la Figure 4, et coordonnée par un Manageur Autonome d’Orchestration (Orchestration Autonomic Manager OAM). L'OAM est assisté par un Manageur Manuel (Manual Manager), c’est-à dire l’utilisateur, et par des manageurs autonomes de niveau inférieur. De leur côté, ceux-ci surveillent et contrôlent les ressources gérées (Managed Resources) par le biais d'une interface de gestion (Manageability Interface). Ces concepts sont introduits principalement dans les grands systèmes informatiques et d'une manière très simple et semi-empirique dans les implémentations existantes du CMGR. En reflétant les architectures autonomes actuellement définies pour les grands réseaux, il apparait intéressant de faire une analogie et d’appliquer ce concept à l'autogestion du MT, plus particulièrement pour la coordination des différentes techniques impliquées dans la résolution de notre problème.

Les terminaux multimodes sont maintenant presque prêts à être déployés dans les véhicules, où ils offriront plus de sécurité, plus d’efficacité dans la gestion du trafic routier et l’accès aux divertissements en ligne. Un ensemble de nouvelles technologies et d’applications est actuellement conçu [70] pour améliorer la qualité de notre expérience en déplacement. Ce nouveau domaine constitue un cas d'application typique de cette thèse, car les nouveaux ordinateurs de bord (On-Board Units, OBUs) ont la capacité de se connecter à au moins trois technologies d'accès, y compris un accès sans fil spécifique dérivé de la technologie Wi-Fi, également appelé technologie ITS (Intelligent Transport Systems) G5, tout en recevant aussi l'information du système de navigation. L'architecture des ITS considère un ensemble varié de terminaux: appareils portables, voitures, camions, véhicules publics tels que les bus, mais aussi les feux de circulation, les panneaux à messages variables ou les centres de surveillance du trafic. En conséquence, les nouvelles applications qu'elle introduit impliquent de nouvelles contraintes sur le sous-système de communications. Par exemple, les applications de sécurité routière mises au point pour prévenir les accidents de voiture, nécessitent des communications inter-véhiculaires avec une latence très faible, ce qui est réalisable principalement avec l'accès ITS G5 en mode V2V (véhicule à véhicule). D'un autre côté, les applications de divertissement peuvent nécessiter une bande passante large ne pouvant être obtenue qu'avec une connexion Wi-Fi ou les réseaux mobiles de nouvelle génération LTE (Long Term Evolution). La sélection du réseau d'accès utilisé dépend donc non seulement du niveau de signal radio, mais aussi des exigences des applications et d'autres paramètres dans le système. Actuellement, la même application utilise toujours le même type de technologie d'accès, quel que soit le contexte du terminal ITS. Le contrôle de chaque périphérique réseau est réalisé avec un logiciel spécifique par la couche réseau. Le terminal étant multimode, chaque nouvelle technologie ou modem nécessite le développement d'extensions du logiciel de contrôle en plus des pilotes de périphériques spécifiques, ce qui permet très peu de flexibilité lors du portage d'un environnement à l'autre. Le monde ITS est donc un cas typique où un algorithme de sélection intelligent de la technologie d'accès, couplé à un fort taux d'abstraction pour la surveillance et le contrôle des différentes technologies et autres dispositifs mobiles, est nécessaire.

Nous estimons que les quatre techniques sélectionnées et analysées ci-dessus peuvent être adaptées et combinées de manière telle à atteindre notre objectif, le schéma visé impliquant aussi une conception inter-couche et des fonctionnalités avancées. L'efficacité des techniques existantes peut être améliorée en les regroupant dans une structure unique. Toutes ces technologies impactent le terminal mobile qui est le seul dispositif que l'utilisateur final peut contrôler. En conséquence, dans la suite de cette étude, une approche innovante a été adoptée, choisissant d'appliquer les modifications au terminal mobile uniquement et laissant le réseau totalement inchangé.

3. Services et Besoins

Trois scénarios types, basés sur cette étude, ont été définis. Leur objectif est d'identifier les cas d'utilisation où le MT ne se comporte pas selon les besoins de son utilisateur. Le premier porte sur la continuité de la session mobile et est illustré à la Figure 5. Il présente un exemple d'utilisation où un utilisateur se déplace depuis un campus vers un réseau cellulaire, pour arriver finalement à un réseau privé connu.

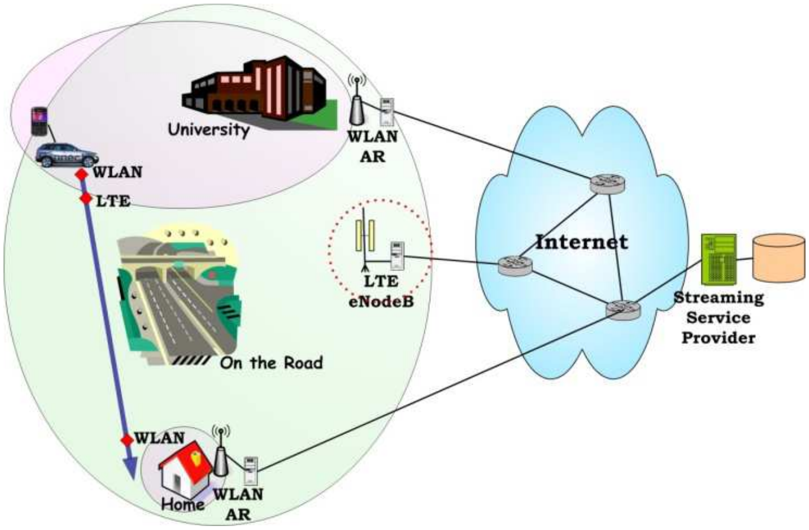


Figure 5: Mobilité entre des opérateurs non-fédérés

Le deuxième scénario concerne les communications véhiculaires. Il montre le terminal ITS dans une voiture qui doit permettre la connectivité à au moins trois réseaux d'accès simultanément et, de plus, recevoir les informations de divers dispositifs tels que le système de localisation ou autres capteurs de surveillance. Le troisième scénario porte sur la notification d'alerte. Il propose un exemple où un utilisateur mobile reçoit une notification d'alerte au public sur l'un de ses réseaux disponibles et lance une application qui nécessite un autre réseau pour recevoir les informations de suivi. Ces situations peuvent impliquer des communications point-à-point (unicast), mais aussi point-à-multipoint (multicast ou diffusion). Ces trois scénarios ont été utilisés comme référence pour l'analyse du comportement du MT au cours de cette étude.

Sur cette base, un ensemble de besoins pour l'architecture cible peut être déterminé. Comme indiqué précédemment, l'approche innovante principale proposée ici est de modifier uniquement le terminal mobile, laissant le réseau totalement inchangé. Cela permet l'itinérance entre des réseaux exploités de façon indépendante. La connectivité doit être conservée de manière efficace tout en restant transparente pour les applications. Le système devra également capitaliser sur une extension du modèle d'abstraction introduit par la norme MIH et sur l'architecture de décision en couches introduite pour les systèmes autonomes, avec une hiérarchie de modules de décision surveillant l'information fournie par des capteurs et coordonnant activement l'action des entités d'exécution, tout en conservant une base de connaissances commune inter-couches.

4. Le Connectivity Control Framework

La Figure 6 montre la chronologie des fonctions impliquées lors de l'utilisation d'un terminal mobile et qui doivent être assurées par la structure cible. Suivant les besoins définis précédemment, le système en couches résultant, la Structure de Contrôle de Connectivité (Connectivity Control Framework, CCF), illustré à la Figure 7, modifie uniquement le MT et laissant le réseau inchangé.

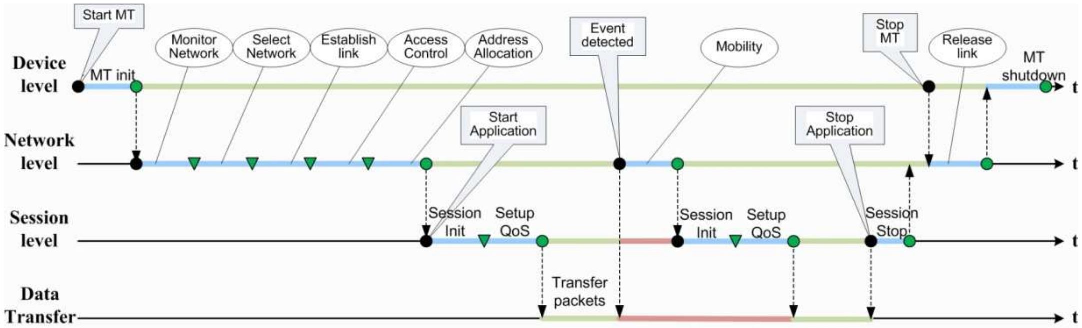


Figure 6: Chronologie de fonctionnement d'un terminal mobile

Certains des composants, identifiés dans la figure par des blocs hachurés, sont présents dans les terminaux existants et demeurent inchangés. Cela comprend les applications, les services réseaux (Networking Services, NS, avec par exemple, les mécanismes existants de handover et de sécurité ou les statistiques de réseau), la pile de protocole TCP/IP (Transmission Control Protocol / Internet Protocol) et les dispositifs internes ou accès sans fil. Le CCF est construit autour de trois grands principes qui garantissent une architecture simple et flexible et qui pourrait se résumer en une simple modification du système d'exploitation du terminal.

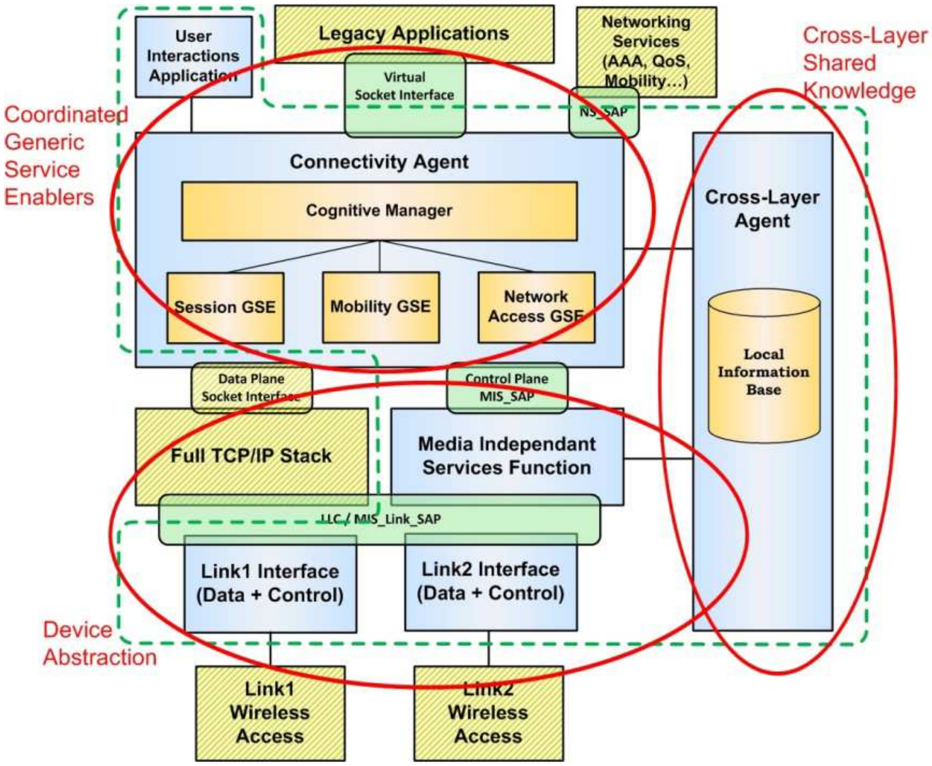


Figure 7: Architecture générale du CCF

Le principe de base est de cacher l'hétérogénéité et la diversité des dispositifs et des réseaux d'accès derrière une interface abstraite qui facilite un éventail de services plus large que la gestion de handover. Ceci est réalisé par la Fonction Services Indépendants du Média (Media Independent Services Function, MISF) et par les interfaces de liaison (Link Interfaces, LI). Un deuxième principe est de partager la connaissance du contexte de terminal et de son environnement entre les différents composants d'une manière inter-couches. Cet objectif est réalisé par l'agent Cross-Layer (CLA) qui enregistre la configuration, les règles et les états dans une base d'information locale (Local Information Base, LIB). Enfin, des facilitateurs de services génériques (Generic Service Enablers, GSE), intégrés dans un agent de connectivité (Connectivity Agent, CA) s'occupent de fonctions de services de base spécifiques. Ils assurent la connectivité transparente et optimisée du terminal et son comportement opérationnel, faisant face aux changements dynamiques et aux événements dans l'environnement réseau, tout en préservant la continuité du transfert des données applicatives. Ceci est réalisé par le GSE de réseau d'accès (Network Access GSE, NAGSE), le GSE de mobilité (Mobility GSE, MGSE) et le GSE de Session (Session GSE, SGSE). Afin de renforcer ces principes et leur efficacité, le Manageur Cognitif (Cognitive Manager, CM) a été introduit dans le CA. Il coordonne les actions des GSE de manière autonome et ne dépend de l'intervention humaine (User Interactions Application, UIA) que lorsque le niveau de confiance du résultat de ses algorithmes d'autogestion est trop faible.

Les composants définis dans le CCF sont basés sur des concepts et technologies existants, les améliorant au besoin. Ils combinent leurs actions individuelles afin d'amener l'ensemble de la structure à son niveau attendu de résilience et d'efficacité.

Le MISF est une couche d'abstraction chargée de gérer la multi-modalité sans fil du terminal. Il s'agit d'un élément clé du système, car il fournit les fonctions pour l'interaction abstraite entre les accès sans fil et autres dispositifs matériels et les couches supérieures, cachant les spécificités individuelles. Il est basé sur les services IEEE 802.21 MIH, mais n'est pas limité au handover. Il fournit un ensemble de services complémentaires, y compris la surveillance des réseaux d'accès disponibles, la récupération de l'état du système et des statistiques, la configuration des ressources avec un certain niveau de qualité de service (Quality of Service, QoS), la gestion des sources d'énergie, du système de localisation, du support multimédia ou l'activation des services multicast et de diffusion.

Au niveau inférieur, les LI font le lien entre le MISF et les pilotes des périphériques. Il y a une LI par périphérique, complètement spécifique de sa réalisation. Sa fonction principale est de traduire les commandes MIS et de les transférer vers le pilote destinataire. Elle agit comme point de terminaison pour la récupération des paramètres vers les couches supérieures, programmant au besoin une surveillance périodique du périphérique. Elle reçoit les primitives MIS de configuration et exécute des procédures internes pour pouvoir remonter les mesures ou les événements souscrits. Son emplacement en bordure du CCF minimise l'énergie globale et la puissance de traitement consommées par la structure.

Les composants LI associés aux dispositifs sans fil contrôlent les technologies d'accès présentes dans le terminal en fonction de leurs contraintes spécifiques. Néanmoins, un terminal mobile, qu'il s'agisse d'un ordinateur portable, d'un smartphone ou d'un OBU, contient d'autres dispositifs, à savoir des systèmes de localisation, des systèmes d'alimentation, ou des étiquettes et capteurs de lumière. De la même manière que pour les interfaces sans fil, ces dispositifs peuvent être commandés et surveillés. Quand ils se rapportent à la connectivité du mobile, ils permettent d'en améliorer la gestion intégrée et coordonnée grâce au CCF et au MISF, à condition qu'un composant LI spécifique soit disponible.

Un ensemble de primitives plus simple que pour le MIH a été défini pour rendre les interactions MIS génériques. Une primitive Link_Action transporte une commande venant des couches supérieures. Dans le sens inverse, le Link_Report indique une information à venant d'une LI. Une primitive Link_Information est utilisée par le MISF pour échanger les valeurs des paramètres avec la LIB, tandis que le Link_Configure enregistre les souscriptions pour des rapports de mesure spécifiques de l'appareil via la LI. Avec ces procédures, le MISF et les LI apportent à la structure la capacité de gérer, de manière abstraite et flexible, les différentes interfaces réseau et les périphériques matériels présents dans un terminal, pour parvenir à une connectivité optimisée.

Mettant en œuvre le second concept du CCF, l'agent Cross-Layer (CLA) recueille les données provenant des dispositifs et des technologies d'accès, ainsi que des couches supérieures, les rassemble dans la base d'information locale (LIB) et les fournit à la demande aux autres composants du CCF.

L'approche cross-layer adoptée ici est un hybride de deux mécanismes cross-layer traditionnels afin de conjuguer leurs avantages. Un moteur cross-layer, le CLA, est introduit pour fonctionner en tant que serveur d'information locale. Il gère une mémoire locale, la LIB, afin de la rendre accessible aux autres composants de la structure, tout en préservant son intégrité. En parallèle, des interactions verticales entre les autres composants sont conservées pour transférer les événements et les commandes. Cela permet de répartir la complexité et garantit une réponse rapide de la structure dans son ensemble aux changements de l'environnement externe.

La LIB est la source de connaissances partagées par l'ensemble du système. Elle contient toutes les données pertinentes pour un fonctionnement optimisé de la structure. Ces données sont classées en trois types: *(i)* des informations prédéfinies, mémorisée lors de la configuration, que ce soit par l'utilisateur ou par des bases de données distantes dans le cloud ou sur les serveurs de l'opérateur de réseau (MIIS / ANDSF), *(ii)* des informations d'état sur le mobile et son environnement rapportés par les autres composants du CCF, *(iii)* des règles et des fonctions d'utilité résultant du processus d'apprentissage. Le CLA gère le processus d'apprentissage dans le CCF. A partir des informations reçues de l'utilisateur et de l'environnement physique via les autres composants, il analyse les valeurs des paramètres et les règles qui ont été appliquées pour déterminer les valeurs des récompenses permettant d'optimiser le fonctionnement du CCF. Avec sa LIB, le CLA apporte à la structure la capacité d'apprendre, d'enregistrer et de distribuer une connaissance commune sur le fonctionnement interne du terminal et son environnement. Cette connaissance est partagée par les différentes couches de la structure.

Les GSE sont les éléments clés de ce système. Ils permettent aux services traditionnels de bénéficier d'une infrastructure agnostique de la technologie. Ils complètent au niveau des services l'abstraction introduite par le MISF. Ces blocs fonctionnels sont appelés génériques car chacun d'eux fournit un ensemble de fonctionnalités spécialisées; ils gèrent les spécificités des applications et des services de réseau existants (NS), et fournissent un moyen unique pour transférer ces services vers les couches inférieures. Ils peuvent interroger la LIB pour obtenir les métriques cross-layer pertinentes et les fournir aux NS. Ils agissent comme des utilisateurs MIS et masquent l'interface MISF aux applications existantes et aux NS.

Le NAGSE traite les aspects liés à la surveillance de la disponibilité des réseaux, l'apprentissage des caractéristiques des accès inconnus et la sélection du meilleur accès grâce à l'exécution de son algorithme sur un ensemble de paramètres extraits du CLA. Pour la découverte et la surveillance des réseaux disponibles, il utilise principalement les informations reçues par les interfaces locales, soit pour déterminer la disponibilité d'un réseau connu, soit pour apprendre les informations système d'un réseau inconnu: sa technologie radio, le nom du réseau, la qualité du signal, ses capacités et la bande passante disponible.

L'algorithme de sélection du réseau d'accès dans le NAGSE peut être invoqué par la fonction de coordination autonome dans le CA à partir de plusieurs états différents du système: lorsque le terminal démarre, quand une nouvelle application est lancée ou lors de l'exécution d'un handover. L'objectif est d'appliquer un algorithme à un ensemble de paramètres et d'en tirer une configuration sous la forme d'une liste ordonnée par préférence des réseaux d'accès. Les critères introduits par les règles régissent les métriques suivantes: une meilleure couverture, la stabilité de la connectivité, l'équilibrage de charge, l'efficacité énergétique, les exigences des applications en termes de bande passante, de QoS, de technologie ou de support réseau, la stabilité de la capacité et la sécurité du réseau.

L'algorithme le plus utilisé, le SAW, est choisi ici pour calculer cette décision, car il est simple, converge en un temps limité et nécessite un temps de traitement réduit. En effet, seule une valeur de score doit être calculée pour chaque réseau d'accès candidat. Les différents scores permettent de déterminer la liste ordonnée de paires (réseau d'accès, score). Le SAW permet d'utiliser un plus grand nombre d'attributs et reflète ainsi davantage le contexte de l'utilisateur et du terminal. De plus, cet algorithme est très flexible car il nécessite seulement que des valeurs de récompense soient allouées aux nouveaux attributs par le CLA, comme décrit précédemment. Grâce à ces fonctionnalités, le NAGSE apporte à la structure la capacité de caractériser l'environnement réseau de manière très précise et de sélectionner le meilleur réseau d'accès pour chaque application, en tenant compte de plusieurs critères indépendants dans un algorithme flexible.

Le MGSE est la partie la plus couramment développée de l'agent de connectivité. Son rôle est de s'occuper des services relatifs à la connectivité, y compris les interfaces réseau et la gestion de lien, la sécurité réseau au niveau L2, la récupération de l'adresse réseau, la réception et le filtrage des événements du réseau et des périphériques, tout en gardant la trace de l'emplacement et des connexions courants. Le MGSE apporte à la structure la possibilité d'être connectée à différents types de réseaux en utilisant des procédés et des mécanismes abstraits. De plus, il filtre de manière intelligente et signale de manière dynamique les changements dans le contexte du MT, qu'ils soient internes ou dans l'environnement réseau externe. Le fonctionnement du MGSE est complètement indépendant des protocoles de mobilité (MIP, PMIP, SIP ou autres) qui peuvent être disponibles dans le réseau et/ou le terminal et font partie des NS.

Le SGSE traite des aspects liés à la gestion des sessions de données ouvertes par les applications. Ce composant effectue la traduction d'adresses entre l'adresse personnelle (PA), vue par l'application, et l'adresse IP locale (LA) vue par le réseau. Pour chaque application exécutée, il existe deux adresses utilisées par le CCF. La PA est attribuée à l'application et maintenue identique tout au long de la session [11]. La LA dépend de l'emplacement réseau de l'utilisateur. C'est l'adresse courante dans le réseau connecté. Pendant les handovers, le SGSE exécute un mécanisme inspiré des techniques de mise en mémoire tampon DTN, stockant les paquets reçus de l'application dans une queue tampon locale jusqu'à ce qu'une nouvelle connexion soit établie de manière définitive. Le SGSE redémarre la connectivité TCP quand elle a été cassée à cause d’un changement de connexion réseau. Lorsqu'une application démarre, il ouvre une socket sur l'interface Socket virtuelle, fournissant l'adresse et le numéro de port de la destination, comme avec une socket de l’API (Application Programming Interface) classique. Puis, le SGSE ouvre une socket réelle sur l’API socket de la pile TCP/IP, avec les mêmes propriétés que la virtuelle, fournissant la LA comme adresse source. Pendant un handover inter-domaine, il résout la rupture de la session en rétablissant automatiquement une nouvelle connexion liée à la nouvelle LA à travers le nouveau réseau d'accès. Dans le cas où TCP est utilisé, il s'assure que la fenêtre de congestion TCP est définie à la même valeur que lors de la fin de la session précédente, avant sa défaillance, afin d'éviter le mécanisme de démarrage lent et de réduire l'impact du handover sur l'application locale. Le SGSE apporte à la structure des mécanismes qui lui permettent de récupérer lorsque le déplacement du terminal a mis en danger le fonctionnement d'une application en cours d'exécution. Il gère toutes les sessions, ouvre et répare les connexions en fonction des besoins, tout en permettant la continuité de la connectivité avec le nœud correspondant.

La Figure 8 illustre comment la hiérarchie de décision des AS a été mappée sur notre structure pour fournir un système autonome en couches.

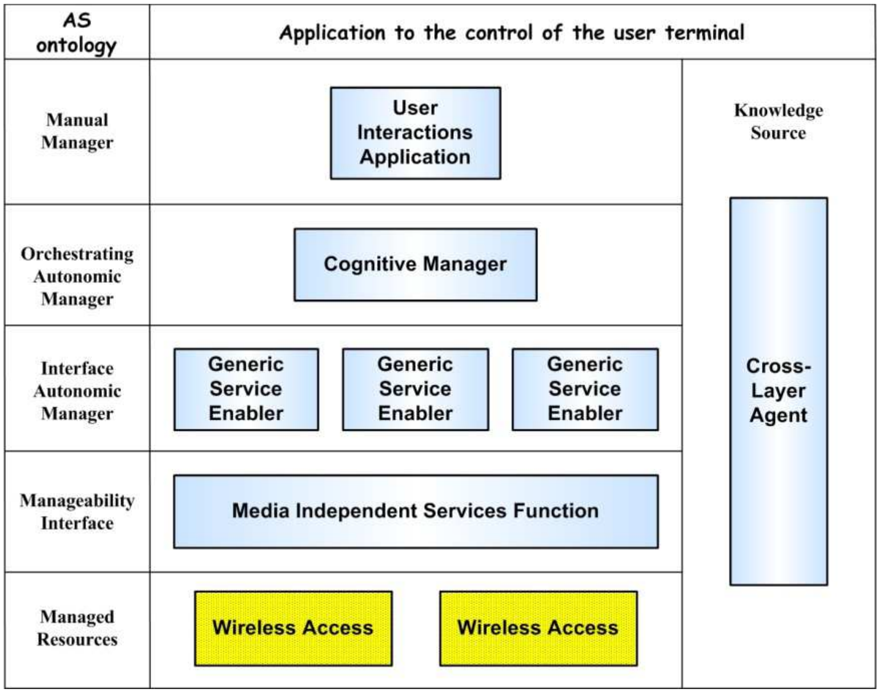


Figure 8: Modèle de décision en couches du CCF

Son étude de bas en haut montre que les technologies d'accès sans fil et les autres périphériques sont les ressources gérées dans le terminal. Les composants LI et le MISF servent d'interface de gestion, pour le suivi et le contrôle des comportements des ressources grâce aux services d’événement et de commande du MIS. Les GSE interagissent directement avec le MISF, chacun d'eux étant dédié à un rôle spécifique. Ils doivent être coordonnés pour fournir un comportement autonome intégré. Le CM joue le rôle de contrôleur de ce système autonome, orchestrant les fonctions d'autogestion du MT pour augmenter le niveau d'efficacité de la structure dans son ensemble. Il est assisté par l'UIA, une interface utilisateur simple pour le propriétaire humain du terminal, qui joue le rôle du Manual Manager dans l'ontologie AS. Le CM obtient ses informations et déclenche les actions à travers les GSEs. Il coordonne leurs actions en fonction de son propre état et des événements reçus, toute décision ou exécution étant générée par le CM. La fonction de source de connaissance est mappée sur le CLA et sa LIB. Le CM apporte à la structure la capacité de fonctionner d'une manière intelligente et autonome en cachant la complexité du maintien de la connectivité réseau à l'utilisateur mobile. L'utilisation du terminal bénéficie ainsi d'une robustesse accrue, d'une meilleure adaptabilité aux événements internes ou externes et d'une efficacité améliorée.

L'UIA établit le lien avec l'utilisateur mobile et lui permet de contrôler le fonctionnement de son mobile en fonction de ses besoins. Il est utilisé pour obtenir les préférences de l'utilisateur au moment de la configuration ou pour valider les décisions du CCF durant son fonctionnement, c'est à dire chaque fois que des informations d'origine humaine sont nécessaires. Une telle interaction est nécessaire et est appelée à devenir moins fastidieuse avec l'apparition d'interfaces plus intuitives basées sur la voix plutôt que sur des écrans pop-ups.

La Figure 9 représente la configuration de réseau utilisée pour l'analyse opérationnelle de la structure CCF. Le domaine A est un réseau WLAN privé sans aucun support de mobilité. Cela peut être le domicile de l'utilisateur ou un hotspot dans un café. Le domaine B est un réseau WLAN de campus équipé d'un support de mobilité, tel que MIP ou PMIP. Le domaine C est un réseau cellulaire de type LTE et utilise la mobilité spécifique au 3GPP seulement.

L'analyse est divisée en scènes, chacune d'entre elles étant un sous-ensemble des scénarios présentés précédemment. La figure montre la position et le mouvement du MT dans chaque scène. La configuration du terminal, son démarrage (MN1a/b) et le lancement d'une application (MN2) se produisent dans les trois scénarios types. Ensuite, trois scènes impliquant la mobilité inter et intra-domaine sont analysées, comme le montrent MN4a, MN4b et MN5a. Elles correspondent au premier scénario, avec le campus de l'université représenté par le domaine B (WLAN avec mobilité), le réseau cellulaire représenté par le domaine C et le réseau personnel de l'utilisateur représenté par le domaine A1 (réseau préféré sans support de mobilité). La dernière scène analyse le cas du troisième scénario, la réception d'alerte d'urgence.

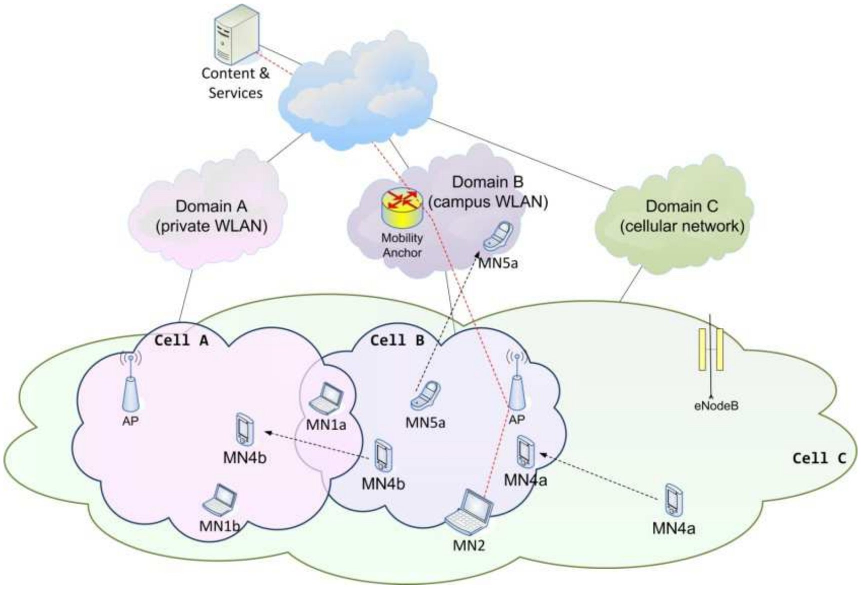


Figure 9: Plan du réseau pour l’analyse opérationnelle

5. Validation du Modèle et Résultats

Cette section a pour but de valider le schéma technologique qui vient d’être défini afin de démontrer les possibilités et les avantages de l'approche envisagée pour la structure. La Figure 10 montre le résultat d'expériences qui ont été effectuées à l'aide d'un terminal réel pour cadrer la question de la continuité de session des applications.

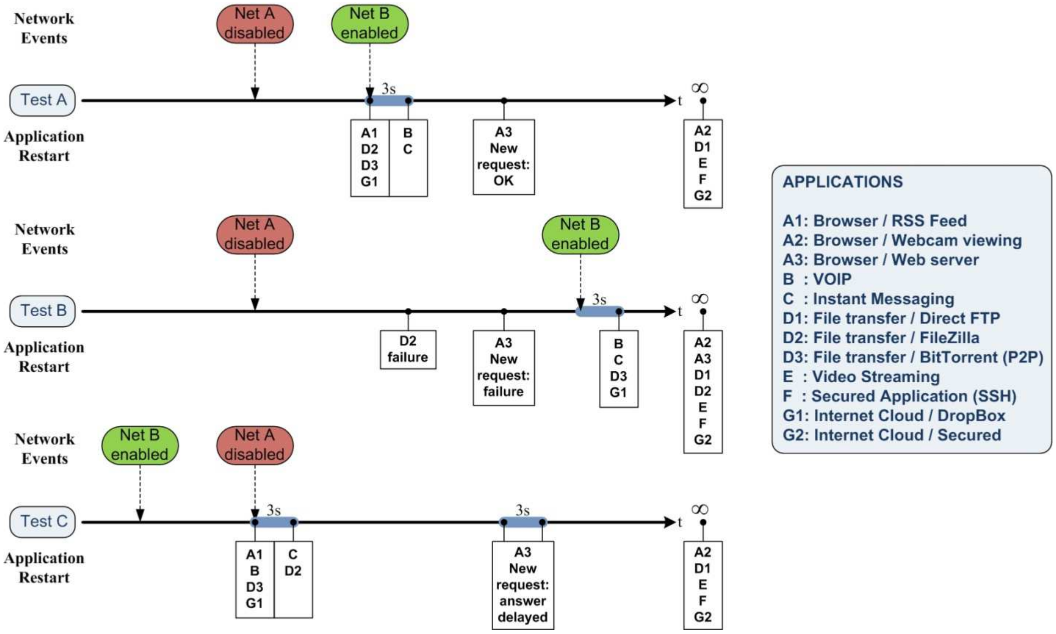


Figure 10: Chronologie de récupération des applications

Ce terminal était équipé de deux interfaces fonctionnant sur des réseaux indépendants. Pour chaque test, l'application a été lancée sur le NetA, puis NetA a été désactivé et NetB a été activé avec un délai variable, comme indiqué sur la figure. Nous avons pu observer que, bien que quelques-unes des applications courantes utilisées soient capables de redémarrer par elles-mêmes, d'autres sont bloquées ou arrêtées, comme le montre le temps de récupération infini sur la droite de la figure.

Cette expérience a pu être reproduite avec un modèle de simulation qui a été développé dans le but de valider le problème et de constituer une référence pour l'évaluation future du système défini. Le résultat de ce test est intégré aux résultats ci-dessous, sous la forme des deux premières colonnes de chaque graphique.

Un modèle de simulation a donc été réalisé qui intègre des nœuds mobiles communiquant dans un environnement hétérogène, comme illustré sur la Figure 11 avec la configuration du réseau utilisé pour la simulation. Le prototype a été développé en utilisant la plateforme OMNET++ [11], un système de simulation à événements discrets.

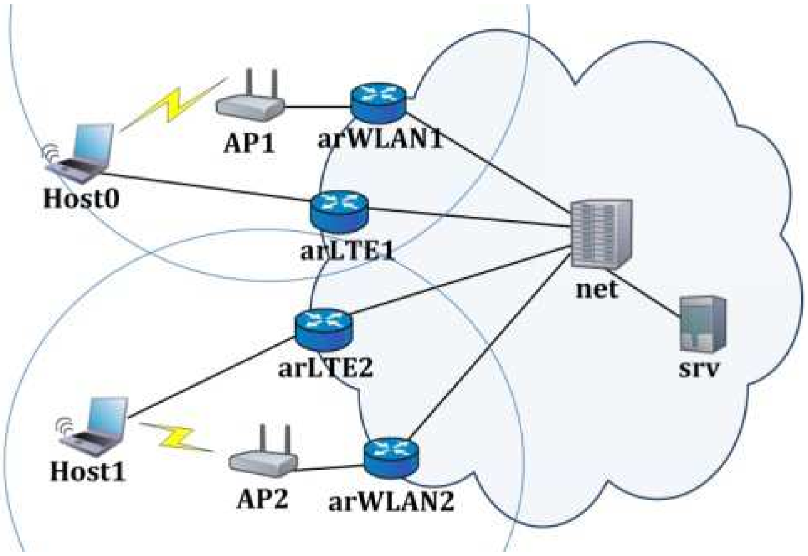
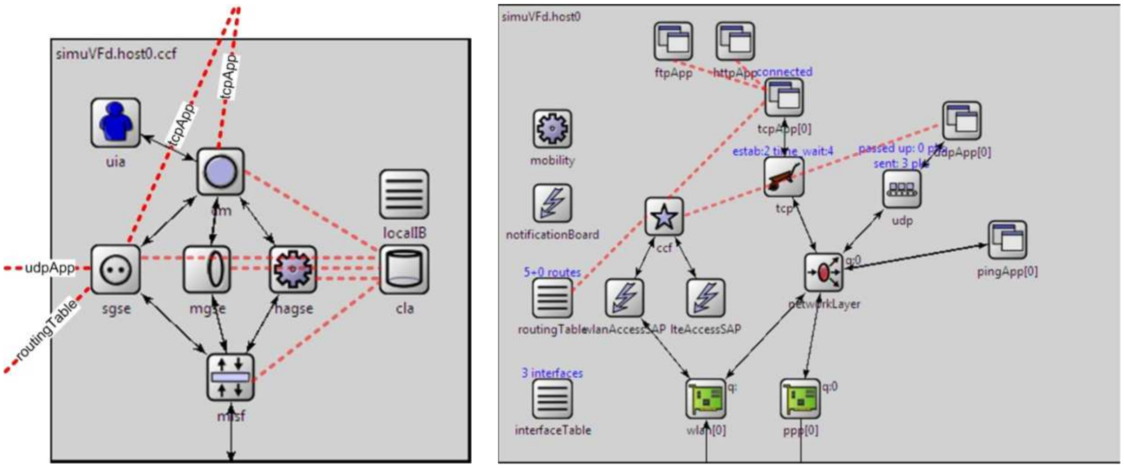


Figure 11: Scénario du réseau simulé

Les expérimentations exécutent des tests de connexion Ping, une application de navigation sur le Web ou une application de transfert de fichiers à partir d'un terminal mobile qui se déplace de façon aléatoire dans le terrain de jeu hétérogène, tandis que le reste des paramètres du modèle restent inchangés. Ce choix permet de mieux comparer les résultats.

La cellule LTE définie a une couverture globale et fournit un accès permanent, tandis que la disponibilité WLAN est limitée à l'intérieur du cercle représenté sur la figure autour de chaque point d'accès. Les résultats de la mise en œuvre du prototype CCF sont comparés à deux autres cas correspondant à l'expérience initiale: terminal avec une seule technologie sans fil (WLAN fixe) et terminal mobile équipé d'un gestionnaire de connexions (CMGR) qui favorise l'accès WLAN, afin d'obtenir la plus grande bande passante possible et de réduire le coût de la communication. Le composant CCF réalisé est strictement conforme à l'architecture décrite dans la section 4 et est représenté dans la Figure 12. Cinquante tests de simulation sont exécutés avec des mouvements aléatoires du MT pour chaque scénario de test afin d'obtenir une plus grande confiance dans l'ensemble des résultats. Chaque test est réalisé avec un temps de simulation d’une durée fixe de 2000s.



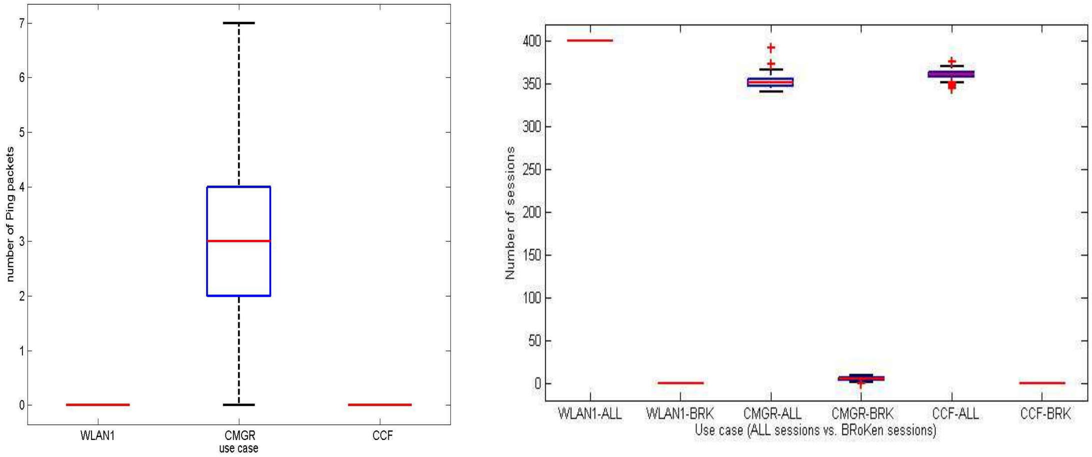
a) Structure CCF

b) Terminal sans fil avec le CCF

Figure 12: Réalisation du CCF pour la simulation

Les principaux critères de réussite sont la minimisation du nombre d'octets de données qui ne sont pas arrivés à destination, le redémarrage des sessions TCP cassées et le temps entre deux handovers pour contrôler l'effet de ping-pong entre les deux réseaux d'accès. Ces critères signalent plus l'efficacité de fonctionnement du MT que les performances du système en termes de débit. Afin de permettre cette évaluation des avantages du CCF, les statistiques suivantes sont enregistrées lors de chaque exécution de la simulation: nombre d'octets transmis et reçus par les applications, nombre de connexions TCP ouvertes / cassées lors de l'essai, nombre de handovers et temps entre deux handovers, temps de connexion sur chaque technologie et au total, utilisation des queues DTN et durée du processus DTN, datage du dernier paquet reçu par l'application dans le mobile.

Les résultats obtenus avec ces simulations sont présentés dans les figures suivantes. La Figure 13a montre les mesures obtenues avec le test de Ping. Avec le WLAN fixe, le terminal reste sous la couverture du réseau sans fil pour l'ensemble de la simulation, donc aucun paquet n’est perdu. Lorsque le terminal se déplace et que le CMGR commute la connectivité, les mesures montrent une perte comprise entre 3 et 7 paquets sur toute la durée de la simulation, avec une moyenne de 3 paquets. Lorsque le CCF remplace le CMGR, le taux de perte retombe à 0%, et devient semblable au cas d’utilisation du WLAN fixe. Ce résultat est dû à l’aptitude du MISF de signaler très tôt le futur évanouissement du réseau, permettant ainsi aux couches supérieures de transférer la connectivité vers une autre liaison disponible avant que l'ancienne ne soit cassée.



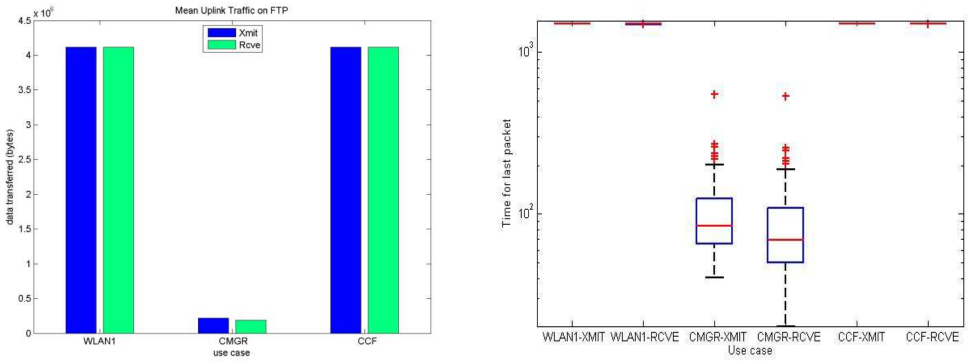
a) pertes sur le test de Ping

b) Session HTTP: Sessions cassées

Figure 13: Comparaison des résultats de simulation lors de l’utilisation du CCF.

La Figure 13b montre le nombre de sessions cassées (-BRK) en fonction du nombre total de sessions ouvertes (-ALL) pour l'application interactive de navigation Web. Avec le WLAN fixe, 400 sessions sont démarrées avec succès et le nombre de sessions cassées est égal à zéro. Avec le CMGR, moins de sessions peuvent être établies et plusieurs sessions sont cassées pendant qu'elles s’exécutent. Lorsque le CCF est utilisé, le nombre de sessions interrompues est réduit à zéro. Toutes les sessions se terminent avec succès. Une très petite quantité de requêtes sont relancées par l'application parce que les paquets sur le lien descendant ont été perdus lors du changement de réseau. Ces résultats confirment que le terminal CCF a pu s'adapter avec succès à des événements de l'environnement.

Les résultats présentés dans la Figure 14 mettent l'accent sur la question décisive de la continuité et de la récupération de la session TCP. La Figure 14a compare le nombre d'octets de données reçus à destination sur la liaison descendante (DL) lors de l'utilisation de l'application de transfert de fichiers à partir du serveur d'applications. Avec le WLAN fixe, l'ensemble du trafic est reçu et le dernier paquet est reçu à la fin de la session. Avec le CMGR, l'application FTP n’est capable de transférer des données que jusqu'au premier handover, ce qui est aussi visible sur la Figure 14b. Ensuite, le minuteur de retransmission TCP expire le nombre de ses essais; la session est interrompue et ne peut pas être récupérée. Avec le CCF, la connexion virtuelle n'est jamais cassée. L'application peut remplir sa tâche jusqu'à la fin et transférer toutes ses données. Certaines d'entre elles nécessitent l'aide supplémentaire de la technique de sauvegarde temporaire dans une queue tampon, comme on le verra ci-dessous.



Session FTP: transfert data DL Session FTP: Datage du dernier paquet

Figure 14: Résultats de simulation pour le transfert de fichiers

La Figure 15 effectue la même comparaison avec l'application de navigation Web. La communication est moins impactée, car l'application démarre en permanence de nouvelles sessions. Lorsqu'une session a échoué, le CCF essaie d'ouvrir une nouvelle connexion TCP. La quantité de trafic transféré est légèrement réduite avec le CMGR et avec le CCF par rapport au WLAN fixe, car l'application est interactive et une plus grande partie du trafic de données est transféré à travers l'accès LTE qui est plus sûr, mais en revanche, offre un débit plus réduit.

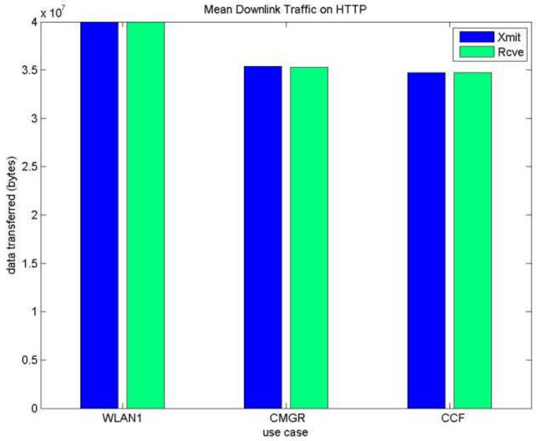


Figure 15: Comparaison du trafic pendant une session Web sur la liaison descendante

La Table 1 montre l'utilisation de la queue tampon dans le SGSE. Quelques paquets sont sauvegardés pendant les handovers («nombre total de paquets en queue tampon») et envoyés lorsque le transfert est terminé, ce qui prévient totalement la perte de paquets. Nous avons pu observer que leur nombre au cours d'un handover ("paquets en queue tampon pendant un HO") reste très faible, et nécessite donc une utilisation de la mémoire à un niveau tout à fait acceptable par rapport à la quantité de mémoire disponible dans les futurs MT.

Table 1: Mesures du mécanisme de queue tampon

Paramètre	FTP	HTTP
Nombre moyen de handovers	15.65	15.47
Temps dans le WLAN (s)	384.33	349.48
Temps dans le LTE (s)	1366.92	1409.07
Nombre total de paquets en queue tampon - max	36.00	10.00
Nombre total de paquets en queue tampon - min	2.00	0.00
Nombre max de paquets en queue tampon pendant un HO	3.00	1.00
Nombre min de paquets en queue tampon pendant un HO	2.00	0.00

Enfin, certaines statistiques ont été enregistrées qui démontrent le faible impact du fonctionnement du CCF sur la puissance de traitement et l'utilisation du mécanisme de queue tampon dans le MT. Un test représentatif a été effectué pour mesurer la quantité d'événements discrets impliquant les composants du CCF et des LI, par rapport au nombre total d'événements pour l’ensemble de la simulation. La Table 2 montre les valeurs mesurées pour cette session de test spécifique. Le nombre d'événements discrets introduits par le CCF reste inférieur à 0,2% du nombre total d'événements.

Table 2: Temps de traitement mesuré

	Nombre d'événements exécutés	Pourcentage
Total de la simulation	761839	
Contribution du *ccf*	409	0.0537%
Contribution des *LIs*	824	0.1082%

Ces tests ont confirmé que le CCF s'exécute conformément à ses objectifs et démontrent les bonnes propriétés de son impact sur les ressources du MT.

6. Application à des Systèmes Réels

L'architecture et les concepts de la structure proposée ont été présentés au chapitre 4. Ils reposent sur trois grands principes: *(i)* l'insertion d'une couche d'abstraction entre les ressources managées et les services de réseau, *(ii)* la fourniture et la coordination des facilitateurs de services génériques, et enfin, *(iii)* le partage de la connaissance à travers l’ensemble de la structure. Ces principes ont été mis en œuvre et évalués lors de la conception de systèmes réels impliquant des terminaux mobiles multimodes.

L'abstraction des spécificités de la technologie a été appliquée au contexte de la mobilité transparente et de la gestion de QoS dans les réseaux hétérogènes, à la géo-diffusion inter-couches et pour améliorer le support de services de transfert de la vidéo dans les futurs réseaux d'opérateurs mobiles. Cette couche d'abstraction a également été améliorée pour inclure de façon détaillée le récent accès sans fil LTE dans la norme MIH. Le principe qui régit les facilitateurs de services génériques a été exploité pour contrôler la diffusion géo-localisée de messages dans l'environnement ITS ou dans le cadre de systèmes d'alerte publics, et pour concevoir la couche de gestion des terminaux, y compris le service de sélection du profil de communications. Enfin, le concept de sauvegarde et de distribution des variables du système à travers les différentes couches grâce au contrôle d'une entité dédiée a été utilisé pour la couche de gestion du terminal ITS.

La diversité des situations d'application possibles pour chacun des grands principes du CCF montre la généralité de l'approche et prouve qu'ils peuvent être bénéfiques par eux-mêmes individuellement. Cependant, dans la plupart des cas, au moins deux d'entre eux ont été combinés et coordonnés, améliorant ainsi de façon évidente l'efficacité du système évalué. Le CCF, qui combine les trois principes, est donc parfaitement en mesure de résoudre le problème posé par la rupture des sessions des applications lors de la traversée de réseaux indépendants non fédérés. Il permet une approche centrée sur l'utilisateur, tout en étant assistée par le réseau, pour la sélection du réseau d'accès le plus approprié.

7. Conclusion et Perspectives

Une structure intégrée cross-layer pour la gestion des interfaces hétérogènes de manière autonome, le CCF, a été proposée dans cette thèse. L'approche adoptée ici est que le CCF est restreint au terminal mobile et n'a aucun impact sur l'infrastructure du réseau, tout en conservant une compatibilité totale avec les normes de communications existantes. Son modèle est basé sur trois grands principes: *(i)* un agent cross-layer qui maintient et partage les connaissances acquises par les autres composants de la structure; *(ii)* une couche d'abstraction qui masque les spécificités du réseau au reste de la structure et comprend de plus le support d'autres périphériques tels que les systèmes de localisation ou divers capteurs, *(iii)* plusieurs facilitateurs de services génériques coordonnés, responsables de tâches spécifiques et gérant les diverses fonctions nécessaires à l'utilisation optimale de la connectivité du terminal.

Pour confirmer ses avantages, un modèle de simulation a été développé, mettant en œuvre l'ensemble de la structure et testant son comportement dans un environnement à base de réseaux sans fil hétérogènes. Alors même que des fonctionnalités supplémentaires ont été installées dans le terminal mobile, l'accroissement de la consommation d'énergie a été limité en plaçant les fonctions d'interrogation périodique et de surveillance des périphériques et réseaux en bordure du système. De plus, les éléments de cette structure ont été appliqués à un large éventail de situations avec des systèmes réels. Ces cas d'application variés démontrent la généralité de l'approche choisie, tout en ouvrant la voie à une acceptation rapide. Le concept qui consiste à restreindre les changements au terminal mobile distribue l'effort dans le système de communications, réduit le risque de goulets d'étranglement dans les fonctions du réseau et améliore son évolutivité. Aucune entité de réseau additionnelle ne doit être déployée et entretenue par les opérateurs, l'installation et la configuration du système sont simplifiées.

Cette étude a examiné en détails la répartition des fonctions et l'intégration et la coordination du système proposé. Pour chaque composant pris individuellement, des solutions existantes satisfaisant aux principaux besoins ont été retenues et les améliorations nécessaires décrites. Une suite future de ce sujet se propose d’effectuer une analyse et une définition plus précises des facilitateurs de services génériques introduits dans le CCF: la sélection de réseau d'accès, la gestion des sessions et le support de la couche réseau, afin d'optimiser les services qu'ils fournissent. Une autre direction consiste à raffiner les modalités d’enregistrement des paramètres de contexte et l'évaluation des valeurs des récompenses pour l'algorithme de décision qui supporte le fonctionnement autonome du mécanisme.

BIBLIOGRAPHY

1. Publications

The results obtained in this dissertation have been published in:

[1] Michelle Wetterwald, Christian Bonnet, Daniel Camara, Sebastien Grazzini, Jérome Fenwick, Xavier Ladjointe, Jean-Louis Fondere, "Future architectures for public warning systems", ICNS 2011, 7th International Conference on Networking and Services, Venice/Mestre, Italy, May 2011

[2] Michelle Wetterwald, Teodor Buburuzan, Gustavo Carneiro, “Enabling IEEE 802.21 in a B3G cellular experimental network”, ICT-MobileSummit 2009, 18th ICT-MobileSummit Conference, Santander, Spain, June 2009

[3] Michelle Wetterwald, "A Case for Using MBMS in Geographical Networking", ITST 2009, 9th International Conference on ITS Telecommunications, Lille, France, October 2009

[4] Daniel Câmara, Christian Bonnet, Michelle Wetterwald, Navid Nikaein, "Multicast and virtual road side units for multi technology alert messages dissemination", WMAPS 2011, 1st International Workshop on Mobile Ad-Hoc Networks for Public Safety Systems, October 21, 2011, Valencia, Spain

[5] Michelle Wetterwald, Fatma Hrizi, Pasquale Cataldi, "Cross-layer Identities Management in ITS Stations", ITST 2010, 10th International Conference on ITS Telecommunications, Kyoto, Japan, November 2010

[6] "Communication Technology Selector", inside D7.1 deliverable, FP7 iTETRIS project, http://www.ict-itetris.eu/, July 2010.

[7] "Spécifications des composants de la couche management", Section 6.5, Deliverable L 2.2.1, SCORE@F project, January 2012

[8] Leonardo Badia, Rui L Aguiar, Albert Banchs, Telemaco Melia, Michelle Wetterwald, Michele Zorzi, "Wireless access architectures for video applications: the approach proposed in the MEDIEVAL project", MediaWiN 2010, IEEE Workshop on multiMedia Applications over Wireless Networks, Riccione, Italy, June 2010

[9] D. Corujo, C. J. Bernardos, T. Melia, M. Wetterwald, L. Badia, and R. L. Aguiar, "Key Function Interfacing for the MEDIEVAL Project Video-Enhancing Architecture", MONAMI 2011, International ICST Conference on Mobile Networks and Management, September 2011, Aveiro, Portugal

[10] Marco Mezzavilla, Michelle Wetterwald, Leonardo Badia, Daniel Corujo, Antonio De La Oliva, "Wireless access mechanisms and architecture definition in the MEDIEVAL project", MediaWin 2011, 6th IEEE Workshop on multiMedia Applications over Wireless Networks, Kerkyra (Corfu), Greece, June 2011

[11] S. Figueiredo, M. Wetterwald, T. Nguyen, L. Eznarriaga, N. Amram, and R. L. Aguiar, "SVC Multicast Video Mobility Support in MEDIEVAL Project", Future Network and Mobile Summit 2012; July 4-6, 2012, Berlin, Germany

[12] Michelle Wetterwald, Christian Bonnet, "Mapping of IEEE 802.21 MIH primitives to EPS/LTE protocols", EURECOM Research report RR-12-265, March 2012, available at http://www.eurecom.fr/en/publication/3701/detail/mapping-of-ieee-802-21-mih-primitives-to-eps-lte-protocols

2. Other Contributions

[13] Michelle Wetterwald, "iTETRIS : The integrated wireless and traffic platform for real-time road traffic management solutions", 2nd ETSI TC ITS Workshop, 10-12 February 2010, Sophia-Antipolis, France

[14] Oscar Gustafsson, Kiarash Amiri, Dennis Andersson, Anton Blad, Christian Bonnet, Joseph R. Cavallaro,. Jeroen Declerck, Antoine Dejonghe, Patrik Eliardsson, Miguel Glasse, Aawatif Hayar, Lieven Hollevoet, Chris Hunter, Madhura Joshi, Florian Kaltenberger, Raymond Knopp, Khanh Le, Zoran Miljanic, Patrick Murphy, Frederik Naessens, Navid Nikaein, Dominique Nussbaum, Renaud Pacalet, Praveen Raghavan, Ashutosh Sabharwal, Onkar Sarode, Predrag Spasojevic, Yang Sun, Hugo

M. Tullberg, Tom Vander Aa, Liesbet Van Der Perre, Michelle Wetterwald, Michael Wu, "Architectures for cognitive radio testbeds and demonstrators - An overview", CROWNCOM 2010, 5th International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 9-11 Juin 2010, Cannes, France

[15] Telemaco Melia, Fabio Giust, Antonio de la Oliva, Carlos Bernardos, Riccardo J Manfrin, Michelle Wetterwald, "IEEE 802.21 and proxy mobile IPv6 : a network controlled mobility solution", Future Network and Mobile Summit 2011, June 2011, Warsaw, Poland

[16] Michelle Wetterwald, Telemaco Melia, Carlos J. Bernardos, "MEDIEVAL : MultimEDia transport for mobIlE Video AppLications", Future Network Technologies Workshop, September 2011, Sophia Antipolis, France

[17] Michelle Wetterwald, "Station Management: Overview on other management functions", CAR 2 CAR Forum 2011, Honda Academy, Erlensee, Germany, November 2011

[18] M. Wetterwald, A. de la Oliva, "Update on Media Specific Mapping for LTE Release 10", IEEE P802.21 Working Group Session #49 Meeting, Hawaii, March 2012, available at https://mentor.ieee.org/802.21/dcn/12/21-12-0027-00-0000-update-on-lte-mapping-for-ieee-802-21.docx

[19] ETSI TS 101 556-1 V1.1.1, “Intelligent Transport Systems (ITS); Infrastructure to Vehicle Communication; Electric Vehicle Charging Spot Notification Specification”, July 2012

3. References

[20] R. Minerva, "Network Paradoxes, Personal Data and the Future Internet", keynote talk at ETSI - Future Network Technologies Workshop, Sophia Antipolis, September 2011

[21] IEEE Std 802.21-2008, “IEEE Standard for Local and Metropolitan Area Networks, Part 21: Media Independent Handover Services”, IEEE, January 2009

[22] Vivek Gupta, "IEEE 802.21 Media Independent Handover, IEEE P802.21 Tutorial", San Diego, CA, July 2006, available at http://www.ieee802.org/21/

[23] E. Piri, K. Pentikousis, “IEEE 802.21”, the Internet Protocol Journal, Volume 12, No.2, June 2009

[24] E. Piri, K. Pentikousis, "Towards a GNU/Linux IEEE 802.21 Implementation", IEEE International Conference on Communications, ICC '09, Dresden, Germany, June 2009

[25] “IEEE Draft Standard for Local and Metropolitan Area Networks: Overview and Architecture”, Draft IEEE P802-REV/D1.2, revision of IEEE Std 802-2001, IEEE, November 2010

[26] H. Schulzrinne and E. Wedland, “Application-layer mobility using SIP”, ACM SIGMOBILE Mobile Computing and Communications Review, July 2000

[27] G. Silvana and H. Schulzrinne, “SIP and 802.21 for Service Mobility and Pro-active Authentication”, Conference on Communication Networks and Services Research (CNSR 2008), May 2008

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

[29] M. Emmelmann, S. Wiethoelter, A. Koepsel, C. Kappler, and A. Wolisz, “Moving Toward Seamless Mobility: State of the Art and Emerging Aspects in Standardization Bodies”, Wireless Personal Communications, vol. 43, no. 3, November 2007

[30] R. Koodli, “Mobile IPv6 Fast Handovers”, RFC 5568, July 2009

[31] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management", RFC 5380, October 2008

[32] M. Ratola, “Which Layer for Mobility? - Comparing Mobile IPv6, HIP and SCTP”, Publications in Telecommunications Software and Multimedia, Helsinki University of Technology, 2004

[33] P. Bertin, Bonjour Servane, J.-M. Bonnin, "A Distributed Dynamic Mobility Management Scheme Designed for Flat IP Architectures", New Technologies, Mobility and Security, NTMS '08. Tangier, Morocco, November 2008

[34] 3GPP TS 23.401, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access"

[35] 3GPP TS 23.402, “Architecture enhancements for non-3GPP accesses”

[36] R. Bolla, R. Rapuzzi, 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

[37] Gong Su and Jason Nieh, "Mobile Communication with Virtual Network Address Translation", Technical Report CUCS-003-02, Department of Computer Science, Columbia University, February 2002

[38] Tom Mahieu, Pierre Verbaeten, Wouter Joosen, “A Session Layer Concept for Overlay Networks” , Wireless Personal Communications, October 2005

[39] V. Cerf, S. Burleigh, A. Hooke , L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-Tolerant Networking Architecture”, RFC 4838, April 2007.

[40] F. Warthman, "Delay-Tolerant Networks (DTNs): A Tutorial; v1.1", Wartham Associates, 2003. Available from http://www.dtnrg.org.

[41] Y. Xia, C. Kiat Yeo, B. Sung Lee, "A Disruption Tolerant Mobility Architecture Towards Convergent Terminal Mobility", 7th IEEE CCNC Conference, Las Vegas, February 2010

[42] Xiaohuan Yan, Y. Ahmet Şekercioğlu, Sathya Narayanan, “A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks”, Computer Networks, Volume 54, Issue 11, August 2010

[43] O. Ormond, P. Perry, J. Murphy, "Network selection decision in wireless heterogeneous networks", IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2005, September 2005

[44] E. Stevens-Navarro, Lin Yuxia, V.W.S. Wong; "An MDP-Based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks", IEEE Transactions on Vehicular Technology, vol.57, no.2, March 2008

[45] Shakeel Ahmad, Hermann Rohling (Dir.), Chunjiang Yin, “Multi-standard Convergence in Mobile Terminals (Master Thesis)”, Hamburg University of Technology, February 2005

[46] Q.-T. Nguyen-Vuong, N. Agoulmine, Y. Ghamri-Doudane, "Terminal-Controlled Mobility Management in Heterogeneous Wireless Networks", IEEE Communications Magazine, vol.45, no.4, April 2007

[47] Chuanxiong Guo, Zihua Guo, Qian Zhang, Wenwu Zhu, "A seamless and proactive end-to-end mobility solution for roaming across heterogeneous wireless networks," Selected Areas in Communications, IEEE Journal on , vol.22, no.5, pp. 834- 848, June 2004

[48] Farooq Bari, Victor C.M. Leung, "Automated network selection in a heterogeneous wireless network environment," IEEE Network , vol.21, no.1, Jan.-Feb. 2007

[49] Phuoc Nguyen Tran, Nadia Boukhatem (Dir.), “Interface selection and flow/interface association decision schemes for multi-interface mobile terminals”, Thesis Telecom ParisTech, September 2010

[50] “Connection Manager specification”, in MEDIEVAL Project, Deliverable D4.3, “Final Specification for mobility components & interfaces”, June 2012

[51] “Open Connection Manager API Requirements; Candidate Version 1.0”, Open Mobile Alliance, June 2012

[52] “Open Connection Manager API Architecture; Candidate Version 1.0”, Open Mobile Alliance, June 2012

[53] http://datatracker.ietf.org/wg/mif/charter/

[54] M. Blanchet, P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, November 2011

[55] M. Wasserman, P. Seite, "Current Practices for Multiple-Interface Hosts", RFC 6419, November 2011

[56] 3GPP TR 23.829, “Local IP Access and Selected IP Traffic Offload”

[57] Wirelesse2e.com, “Traffic Offloading: WiFi to Rescue”, White paper, September 2010; available at

http://wirelesse2e.wordpress.com/

[58] Eugen Borcoci, Mihai Stanciu, Dragos Niculescu, Daniel Negru, George Xilouris, “Connectivity Services Management in Multi-domain Content-Aware Networks for Multimedia Applications”; Proc. of. INTERNET 2011, Luxembourg, June 2011.

[59] Meriem Kassar, Brigitte Kervella, Guy Pujolle, “An overview of vertical handover decision strategies in heterogeneous wireless networks”, Computer Communications, Volume 31, Issue 10, June 2008

[60] Haleh Tabrizi, Golnaz Farhadi, John Cioffi, "Dynamic Handoff Decision in Heterogeneous Wireless Systems: Q-Learning Approach", IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 2012

[61] Lusheng WANG, Daniel Kofman (Dir.), “Sélection de Réseau dans les Réseaux Sans Fil Hétérogènes”, Thesis Telecom ParisTech, January 2010

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

[63] Haffiz Shuaib, Richard J Antony, Mariusz Pelc, “A framework for Certifying Autonomic Computing Systems”, ICAS 2011, Venice, Italy, May 2011,

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

[65] J. Mitola III and G. Q. Maguire, Jr., "Cognitive radio: making software radios more personal," IEEE Personal Communications Magazine, vol. 6, nr. 4, August 1999

[66] S. Haykin, "Cognitive Radio: Brain-empowered Wireless Communications", IEEE Journal on Selected Areas of Communications, vol. 23, nr. 2, February 2005

[67] UniverSelf Project, "UMF Specifications", Deliverable D2.1, June 2011, http://www.univerself-project.eu/

[68] T. Farnham, M. Sooriyabandara, "Cognitive Resource Management for Wireless LAN within the Home", ICT-MobileSummit 2009, 18th ICT-MobileSummit Conference, Santander, Spain, June 2009

[69] SeVeCom project : http://www.sevecom.org

[70] European ITS Communication Architecture Overall Framework, COMeSafety project: http://www.comesafety.org

[71] ETSI ITS Technical Committee, http://www.etsi.org/WebSite/Technologies/IntelligentTransportSystems.aspx

[72] “CAR 2 CAR Communication Consortium Manifesto”, August 2007, available at http://www.car-to-car.org/fileadmin/downloads/C2C-CC_manifesto_v1.1.pdf

[73] ETSI EN 302 665 V1.0.0, "Intelligent Transport Systems (ITS); Communications Architecture", March 2010

[74] SCORE@F (Système Coopératif Routier Expérimental @ France); http://www.scoref.fr/

[75] 3GPP TS 23.246, “MBMS; Architecture and functional description, Release 6”

[76] 3GPP TR 22.968, “Study for requirements for a Public Warning System (PWS) service”

[77] T. Narten, E. Nordmark, W. Simpson, H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[78] MEDIEVAL, Deliverable D3.1, "Concepts for Wireless Access in Relation to Cross-Layer Optimisation", June 2011, available at http://www.ict-medieval.eu

[79] IEEE Std 802.2-1998, “IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical Link Control”

[80] J.D. Martinez-Morales, U. Pineda-Rico, E. Stevens-Navarro , "Performance comparison between MADM algorithms for vertical handoff in 4G networks," 7th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE) 2010, September 2010

[81] S. Chakraborty, C-H. Yeh, “A simulation based comparative study of normalization procedures in multiattribute decision making”, 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu, Greece, February 2007

[82] http://www.omnetpp.org/

[83] A. de la Oliva, T. Melia, A. Vidal, C. Bernardos, I. Soto, A. Banchs, "A case study: IEEE 802.21 enabled mobile terminals for optimised WLAN/3G handovers," ACM Mobile Computing and Communications Review (MC2R), vol. 11, no. 2, April 2007.

[84] T. Melia, A. de la Oliva, I. Soto, C.J. Bernardos, A. Vidal, "Analysis of the Effect of Mobile Terminal Speed on WLAN/3G Vertical Handovers", IEEE Global Telecommunications Conference, GLOBECOM '06, Nov. 2006

[85] "AT&T goes live with LTE; early analyst review praises downlink speeds", available at http://www.mobilebusinessbriefing.com/articles/at-t-goes-live-with-lte-early-analyst-review-praises-downlink-speeds/17315/

[86] The EU IST project Daidalos: http://www.ist-daidalos.org/

[87] http://www.openairinterface.org/

[88] Telemaco Melia, Marco Liebsch, Pablo Serrano and Albert Banchs, "The Daidalos Architecture for Mobility and QoS", WTC, May 2006

[89] IETF internet-draft, draft-ietf-mext-nemo-ro-automotive-req-02, "Automotive Industry Requirements for NEMO Route Optimization", January 2009, work in progress

[90] The EU IST project iTetris: http://ict-itetris.eu/index.htm

[91] Michelle WETTERWALD “Contribution to Facilities Communication Management”, Document ITSWG1(10)0082, Jan 2010

[92] 3GPP TS 36.331, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification"

[93] 3GPP TS 24.301, “Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3"

[94] 3GPP TS 36.300, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2"

[95] 3GPP TS 27.007, "AT command set for User Equipment (UE)"

4. Other Contributions

[96] Michelle Wetterwald, Fethi Filali, Christian Bonnet, Dominique Nussbaum, Lionel Gauthier, Aawatif Menouni Hayar, "A flexible framework for the support of heterogeneous wireless networks", IST Summit 2006, 15th IST Mobile & Wireless Communications Summit, Mykonos, Greece, 4-8 June 2006