Generic Open Link-Layer API for Unified Media Access—GOLLUM

D4.2

"Use of ULLA at operator level and application interfaces"

Contractual date of delivery to the CEC:	**28th February 2006**
Actual date of delivery to the CEC:	**28th February 2006**
Editor:	**A. Gefflaut (EMIC)**
Authors:	**A. Gefflaut, L. Latour (EMIC), F. Negredo (TID), M. Wellens, J. Riihijärvi, P. Mähönen (RWTH), T. Farnham, M. Sooriyabandara (TREL), D. Melpignano (STM), J. A. Galache, V. Gutiérrez (UC), A. van Rooijen (MA)**
Participants:	**EMIC, TID, RWTH, TREL, STM, UC, MA**
Workpackage	**WP4**
Security:	**Public**
Nature:	**R**
Version:	**1.0**
Total number of pages:	**36**

Abstract:

This document describes how the Unified Link Layer API could be used by operators both on terminal equipment and on Radio Access equipment to provide solutions to problems such as efficient and transparent roaming between different technologies, and management of network equipments.

Keyword list: Cellular, GPRS, UMTS, Link Layer Adapter, ULLA, Link, Link Provider, operator level, media independent handover, IEEE 802.21.

Document revision history

Version	Date	Author	Summary of main changes
0.1	20/01/06	EMIC	Initiated the document and suggested table of contents.
0.2	31/01/06	RWTH	Added contribution about usage of ULLA for IEEE 802.21-based vertical handover. Smaller editorial changes.
0.3	22/02/06	TREL	Added contributions on ULLA, OMA and device management
0.8	23/02/2006	EMIC	Integrate contributions from different partners.
0.9	26/02/2006	EMIC	First complete version of the document.
1.0	28/02/2006	RWTH	Finalizing document

Contents

Table of figures

1. Introduction

The goal of the Unified Link Layer API (ULLA) is to provide an Application Programming Interface simplifying access to link layer information. In particular, its goal is to propose a uniform way to access link information, and this, regardless of the technology or the platform used. The proposed API supports querying of information from different types of links, as well as changing their configuration. Additionally, the ULLA supports an event based programming model by allowing programmers to set up criteria for triggering asynchronous notifications bases on link characteristics.

Through several prototype demonstrators, the GOLLUM consortium has demonstrated the usage of the ULLA in different domains such as dynamic adaptations to channel conditions, sensor network routing algorithms and connection management on mobile devices. The prototypes as well that the proposed approach was valid for different types of hardware (sensors, PDA, embedded systems…) or software platforms (Windows CE, Windows XP, Linux, TinyOS).

In this document we concentrate on the usage of the ULLA by network operators to simplify their work or provide new types of services. Operators own both terminal equipments, connected to networks, and the networks themselves. We believe that the ULLA could be used both on terminal equipments and in the Radio Access Network. The following sections give an overview of the possible usage of the ULLA in this environment.

2. Definitions

This section presents a short definition of the most frequently used terms when talking about the ULLA.

Command

Commands are issued from LUs to the ULLA and forwarded to the LPs or Links as is. They support for configuring and controlling the behaviour or state of the addressed entity. Commands and attributes are organized in separate inheritance hierarchies.

Most relevant commands on LPs support the possibility to discover new Links. Links generally might not have so many commands, but for instance connect and disconnect are mandatory. Connectionless links might not do anything there except returning OK.

Events

Events are passed from the LP or Link to the UEP. If the ULLA Core has registered its interest in the update of a particular attribute using the requestUpdate() function, the LP or Link reports either periodically or upon disposal of new measurement values.

Some attributes have to be updated by the LP or Link based on asynchronous events. For instance the Frame Error Rate could be updated when receiving a new frame. Other attributes are evaluated at periodic times, such as the current throughput (bytes within last period).

Link

LLAs can register links for pre-registered link providers to ULLA. For ULLA a link is exhaustively defined by a link class description.

With respect to the ISO/OSI or TCI/IP model a link is intended to be an interconnecting communication channel in layer-2. Pair of layer-2 addresses, which usually identify the two respective peers, uniquely identify each link. In this context, peers are the terminal installations, which implement the layer-2 communication protocol For instance, a single network interface card (NIC) might be able to maintain multiple links with different peers.

Link class description

A link class description exhaustively characterizes the type of link with a set of link properties, also called attributes, and the available commands that can be issued on that type of link.

Some interesting properties of a link are connection oriented (infrastructure) versus connectionless (broadcast or ad-hoc) mode, security mode, and link quality parameters. A common command for links is connect().

Link Layer Adapter (LLA)

Link layer adapters are not visible to ULLA or link users.

A LLA is a piece of software, which registers link providers and links to ULLA by the means of the defined interface. A LLA is typically tailored for a specific device driver or device driver tool, respectively.

Link Provider (LP)

LLAs can register Link Provider with ULLA, defined by their Link Provider class description. A Link Provider itself hosts links with the same link class description.

A Link Provider is an abstraction of a network interface card (NIC). For instance a LLA for a multi-mode NIC supporting GPRS and WLAN might register two link providers with ULLA. Each of them is defined by the means of a specific class description.

Link Provider class description

A Link Provider class description characterizes the type of Link Provider with a set of Link Provider properties, also called attributes, and the available commands that can be issued on that type of link provider.

For instance Link Provider properties can capture power modes of the respective NIC. Commands may set the respective NIC mode in operation, which prevents communication on the hosted links, such as scanning()command might do.

Link User (LU)

A Link User is any application registering with ULLA as a link user.

Here, the term application is understood in a software architectural sense. It does not limit the potential link users to "layer-7" applications with reference to a communication model, but also includes communication middleware, transport entities or routing agents.

Notification

A notification is send from ULLA to the LU when a pre-registered UQL condition on LP or Link attributes is met. Additionally, they can indicate a change in LP or Link state, such as new LP registered or Link deregistered. The UQL language is also used at registration time to specify the associated information coming alongside with the notification.

This ULLA feature is very powerful for LUs that are interested in link adaptation. Most applications might reside on highly portable attribute conditions for many links, such as browsers, while others make use of the full flexibility of the UQL, such as connection managers or multimedia clients.

Query

A query is send from an LU to ULLA in order to retrieve information about LP or Link attributes by the means of a UQL statement. When used with requestInfo() the result is returned as fast as possible and may be empty. When used with requestNotification() the functional call blocks until the condition is met and the result is non-empty.

A query specifies a list of interesting attributes for the LU, such as bandwidth, and optionally a condition that all in the result listed LPs or Links have to meet. The scope of a query can be scaled down to a particular class, such as Cellular-Link-class or IEEE 802.11-LinkProvider-class. In sub-classes more attributes are available, for instance cellular-type of links specific attributes. If an attribute is requested from a class, which does not support it, the query fails. A freshness parameter indicates the requirements on the age of the addressed attributes.

Sub-link:

If a link can be divided into a set of links with the same layer-2 addresses each instance is referred to as a sub-link.

For instance, sub-links might capture the properties of a TDMA system assigning distinct timeslots to different sub-links.

ULLA Core (UC)

ULLA Core is the module that implements all the functionalities of ULLA including the management of the interfaces towards Link Users and Link Providers. ULLA Core comprised of three functional components that deals with command processing, query processing and event processing.

ULLA Storage

ULLA keeps persistent information in the ULLA storage. This can rely on a legacy database or can be a custom realization tuned for a given platform.

The ULLA storage system will typically contain link related information and parameters which will be accessed by the Link User by means of the ULLA Query Language. If necessary, ULLA queries can be translated into an implementation specific query language by the ULLA Core.

Unified Link Layer API (ULLA)

ULLA is an abstraction of an operating-system independent implementation of the defined standard interfaces, which link users and LLAs can use. Generally, the word ULLA can refer to API as well as the system that realises the API.

3. ULLA Description

3.1 ULLA Architecture

The aim of ULLA (Universal Link Layer API) is to provide an API for accessing link-layer functionality and information in a technology independent manner. It will hide network standards heterogeneity behind a standard set of functionalities applicable to the wide range of network technologies. The ULLA provides an abstraction from specific link technologies to the applications or other link users (LUs) by regarding a link to be a generic means of providing a communication service. In this context, links are made available and configured through link providers (LPs) to permit abstraction from specific platforms and technologies. Link users that benefit from ULLA services include, but are not limited to, any higher layer protocols, middleware or application software. Figure 3-1 presents a graphical overview of the different components involved in the implementation of the ULLA.

In the center of the picture, the ULLA core is represented. It contains 3 main components. The UllaQueryProcessing module is in charge of analyzing the queries and notification requests coming from Link Users. The UllaCommandProcessing module handles commands and forwards them to the corresponding Link Provider. The UllaEventProcessing module takes care of handling events arriving from the Link providers (new link arrivals, new value for a link characteristic…) and in particular of the evaluation of registered notification requests. Finally the UllaStorage, represented outside of the Ulla Core, is an optional component used to cache link characteristics collected from the Links and Link Providers in order to avoid access to the drivers or hardware for each query.

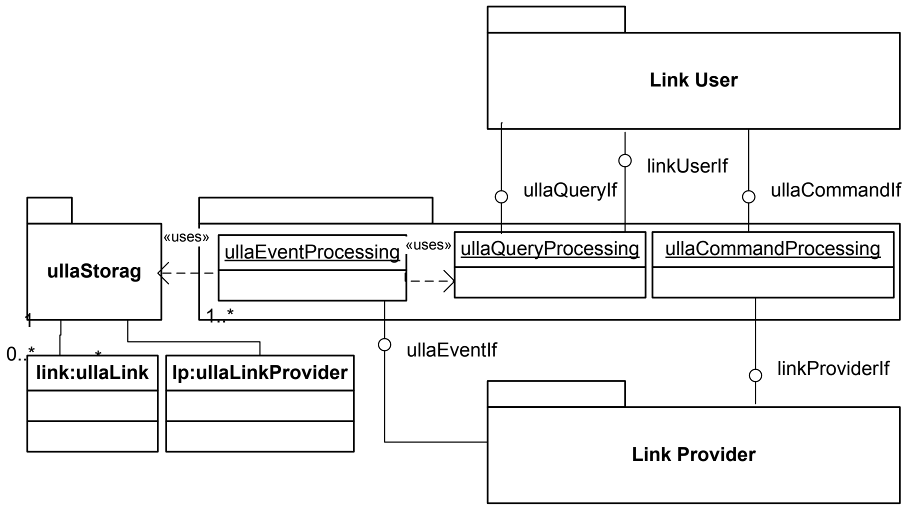

Figure 3-1: The ULLA Architecture.

The main services provided through the ULLA are:

- **Queries**: The ULLA provides a generic querying mechanism allowing applications to retrieve link information in a technologically independent fashion. To enable this, a query language called ULLA Query Language (UQL), a subset of the well-known SQL, is used.
- **Commands**: The ULLA supports a mechanism that allows applications to configure and manage links in a standard way through the use of commands than can be called from user level applications.

- **Events**: The ULLA provides support for asynchronous notifications based on user defined link criteria. Link users can define and register any type of conditions triggering an asynchronous notification through the UQL. Once registered these conditions are dynamically evaluated by the ULLA Core based on events reported by the Link Providers (LP). For example it is trivial with the ULLA to enable a notification when a specific link received signal strength goes under a certain threshold.

ULLA provides an abstract view of link layers to ULLA clients (link users) to facilitate a uniform way to access the wide range of existing link layers, independently of their implementation. In order to manipulate these abstractions, a specific query language (i.e. ULLA Query Language, UQL) is used. A ULLA query specifies a request made by an application to retrieve information about a link layer synchronously (i.e. information request) or asynchronously (i.e. request notification). Finally a command is a request specifying an action that should be executed to modify a link layer state. An overview of the basic elements composing the ULLA architecture is presented below.

UQL Query	Meaning
SELECT linkId, rxBitRate FROM UllaLink;	Retrieve the link identifier and the received bit rate from all available links.
SELECT linkId FROM UllaLink WHERE rxBitRate> 100000;	Retrieve the link identifier of all links having a received bit rate over 100000 bits per second.
SELECT linkId FROM UllaLink WHERE (rxBitRate > 100000) AND (rxQuality > 60);	Retrieve the link identifier of all links having a received bit rate over 100000 bits per second and a received signal quality over 60 percent.

Table 3-1: Example of UQL queries.

Table 3-1 presents 3 simple query examples and their associated meaning. The same UQL is used to register a notification. In this case, the query is kept inside the Ulla Core in order to be re-evaluated later on. Note that the language is both simple and expressive enough to enable any type of query or notification to be defined.

3.2 *ULLA Interfaces & functionality*

ULLA provides a set of unique and well defined functions used by the Link Users (LUs) to access relevant information and issue commands to Link Providers (LPs). It is worth noting that ULLA currently defines interfaces with both LUs and LPs and core functionality (namely, command processing, event processing and query processing) (see Figure 3-1) in order to achieve the above objectives. This kind of separation makes the distinction between different software domains, which are uncorrelated with each other, clear: one domain is intended for application developers, who are willing to control wireless adapters as a whole with a very high level of abstraction. The second domain includes device manufacturers who are more interested in providing a ULLA compatible software layer (as an LLA – Link Layer Adaptor) over their technology and OS specific device drivers. In the LLA context, the API definition is more tightly coupled with the concept of an LP mapped to a physical network adapter.

The interfaces provide functionalities that can be logically split into the following:

1) Sending commands to the ULLA Core (either synchronously or asynchronously)

2) Request notifications from the ULLA Core

3) Querying the ULLA storage system for retrieving information.

Commands directed to the core aim to satisfy some high level application requirements (i.e. bandwidth, delay, link quality etc.). Such commands may be sent in a synchronous or asynchronous way in order to accommodate every application need and design. In particular, asynchronous commands are suited to long running tasks such as the request for wireless LAN infrastructure scanning.

LU interfaces offer the means to notify the application when certain conditions are met. Such conditions are specified via UQL, whereas asynchronous notifications are realised by means of a callback mechanism: the LU provides ULLA with the implementation of a well prototyped function which is called by ULLA upon satisfaction of a registered request. This relieves the application from needing to resort to polling mechanisms, which are inefficient and consequently have high resource consumption.

ULLA supports a concept called role management to handle LUs with different privileges. Through role management, LU's ability to use a set of services provided by ULLA is controlled based on the role and privileges associated with it.

3.3 *Link User Interface Usage*

In order for an LU to use the ULLA it needs to register itself with the ULLA Core. The function *registerLu* will be used to pass a *LuDescr* structure (containing name, description and version) to ULLA. An LU must call the *unregisterLu* method when it stops using ULLA functionality.

3.3.1 Command Handling

The LU has the ability to send commands to the LPs and Links via the ULLA Core. The *doCmd* method is used to execute a synchronous command. The LU needs to pass a *cmdDescr* command description structure. This structure contains the name and parameters of the command to be executed. It also passes the ID of the LP and Link that has to execute the command.

The *requestCmd* method is used to execute an asynchronous command. In addition to the previous parameters there is also a need to pass command descriptor *requestCmdDescr*. This structure contains a pointer to a callback routine to handle the result of an asynchronous command. In order to execute a command periodically a count and a time interval can be set in the *requestCmdDescr* structure. The function will return a unique *cmdId* upon successful queuing of the command. The LU can also cancel a queued command by using the *cancelCmd* method by passing *cmdId* as parameter.

In case several LUs issue conflicting commands to the same link, an arbiter called a Link Manager (LM) can apply suitable policies to resolve conflicts. An interface is defined in the current design to ease the insertion of a third-party LM in the ULLA SW architecture.

3.3.2 Query Processing

The LU can retrieve information from the ULLA Core or LPs by doing an information request. This is done with the *requestInfo* method. With the *requestInfo* method, the LU needs to pass a query string and a pointer to an *ullaResult* handle. The data in the result handle can be retrieved using special *ullaResult* assessor functions. After all the data has been retrieved the LU needs to call the *ullaResultFree* method to free the memory. Sometimes it is necessary to be informed when a certain attribute of a Link is changed. This can be done by requesting a notification. This works in a similar way to the *requestCmd* method and the *requestNotification* method is being used for this. In addition to these parameters, a *RequestNotificationDescr* structure has to be passed, which contains a pointer to a callback function, a count and period. The function will return a *requestNotificationId* when the notification has successfully been queued. This notification request can be cancelled by the *cancelNotification* method.

3.4 Link Provider Interface

This interface provides the means to notify link events received from the LP to ULLA and to pass commands received from LUs to LPs. When registering using the *registerLp* method, an LP needs to pass the structure *LpDescr*. This structure contains information about the LP and also a pointer to its interface. Upon successful registration the ULLA Core returns a unique identifier (*lpId*). LP registration does not automatically imply that there are also active links. The ULLA Core needs to do an active scan to find all available active links. When an LP is unloaded it must call the *unregisterLp* method. It needs to pass the *lpId* it received when registering.

3.4.1 Command Handling

Commands are passed (by the ULLA Core) from the LU to the LP. The *execCmd* method is used to pass a synchronous command to the LP along with the link that has to execute the command. The ULLA Core uses the *lpId* in the *cmdDescr* to address the appropriate LP. The *cmdDescr* contains all other information the LP needs. In the current API definition, the LP has no special function to pass an asynchronous command. The handling of asynchronous commands is completely handled by the ULLA Core. From the LPs point of view the *execCmd* method is used in both cases.

3.4.2 Query Handling

The ULLA Core will normally first retrieve data from the ULLA storage. When newer information than that which is stored is needed, ULLA Core can retrieve this information by using the *getAttribute* method in order to fetch the current value of an attribute from an LP. As parameters, ULLA Core needs to pass the *linkId*, the identifier of the attribute, a data qualifier and a pointer to where the result has to be stored. With the data qualifier the LP can tell if the value should be a HARDCODED, THEORETICAL, ESTIMATED or EXACT value. The memory to store the attribute is allocated by the LP. This memory then has to be freed by the ULLA Core with the *freeAttribute* method when no longer required.

When the LU requests a notification, e.g. when a certain attribute has changed, the ULLA Core will call *requestUpdate*. The ULLA Core needs to supply an event handler to process all the update events. With the *RequestUpdateDescr* structure, information regarding the requested attribute, the count and time interval the update needs to be sent is passed to the LP. The update request can be cancelled with the *cancelUpdate* method. In order to track all different update requests sent to the LPs, ULLA will generate a unique ID for all requests.

3.5 Status

3.5.1 ULLA API definition

The GOLLUM consortium has defined the first version of the ULLA API and the concept has been successfully demonstrated on a number of different platforms with various device and

Operating System capabilities. The final API definition will be publicly available by the end of August 2006.

3.5.2 Implementation

To demonstrate the feasibility of the approach proposed by the ULLA and its applicability to a wide range of software and hardware platforms, the GOLLUM consortium has developed several prototypes. At this point, prototypes have been written for Linux, Windows XP, Windows CE and TinyOS. From a hardware point of view, the platforms used vary from desktop/laptop class computers to PDA and phones and down to sensors.

4. Current trends in the mobile world

Faced with the development of Voice over IP (VoIP) and the quick penetration of WiFi technology, even in public spaces, operators have now realized that there is an urgent need to provide data services based on the Internet Protocol. The following sections describe the different efforts that have been done to integrate cellular with different forms of wireless networks like WiFi, Bluetooth and/or WiMAX.

In this section we present the most important trends in the world of network operators. We limit ourselves to technologies that are somehow directly related to the ULLA.

4.1 Integration of Cellular and Wireless Networks

Established in 1998, the 3rd Generation Partnership Project (3GPP) is a collaboration agreement between a number of telecommunications standards bodies. The scope of 3GPP is to define the evolution of GSM and the radio access technologies that they support.

The 3GPP has studied the internetworking between the 3GPP system and WLAN networks, so that any WLAN technology could be considered as an added access network for a 3GPP network. For the complete integration, 6 scenarios have been considered, in which the internetworking between both technologies is rolled out progressively. The considered are the following:

1. *Common Billing and Customer Relations.* The customer receives a unique bill for the usage of WLAN and 3GPP services.

2. *Authentication, Authorisation, and Tariffication based on 3GPP.* Access control, authorisation, accounting and tariffication are managed by the 3GPP system. Whether the user is operating in the operator's WLAN or roaming at another operator's WLAN. New entities have been created to manage the AAA (Authentication, Authorization and Accounting) which connects with the HSS for AAA and billing functions.

3. *Access to 3GPP Packet Switched Services.* The customer connected through the WLAN uses the PS services offered by the 3GPP operator (e.g. IMS, APNs, etc). Nevertheless, service continuity between both 3GPP system and WLAN is not required. The customer could be connected to the operators WLAN or roaming in another operator's WLAN (visited WLAN). The WAG (WLAN Access Gateway) and PDG (Packet Data Gateway) entities have been specified in order to provide access to the PS 3GPP services. These entities have similar functions to SGGNs and GGSNs in relation with routing and user data transfer from and to IP networks.

4. *Service Continuity.* Services enabled in the previous scenario have to survive when switching between 3GPP to/from WLAN. Session should not be interrupted, but the quality of service could change due to the systems transition.

5. *Transparent Services.* Transparent service continuity is granted between different access technologies. Further data loss is minimized and gap of time during the handover between access technologies.

6. *Access to 3GPP Circuit Switched Services.* Users can have access to the services provided by the Circuit Switched 3GPP domain through the WLAN and switch different access technologies in a transparent way for CS services.

Scenario 1 does not require any change for the 3GPP specifications, whilst scenarios 2 and 3 are included in *release 6*. Roaming is assured from scenario 2, where the 3GPP system manages the AAA functions, but functional roaming only becomes real from scenario 3 where data services are accessible even from a visited WLAN.

The following sections present different solutions that try to integrate the usage of wireless data networks (most of the time WiFi) with existing cellular networks (scenarios 4, 5 and 6)

4.1.1 The UMA Approach

UMA which stands for Unlicensed Mobile Access is a technology developed to provide access to GSM and GPRS mobile services through unlicensed spectrum technologies such as WLAN and Bluetooth. Initially promoted by an independent group of companies, UMA is now included in 3GPP *release 6* where it is known as GAN (Generic Access Network). By deploying UMA, existing WLANs could be turned into a seamless extension of a mobile network allowing users to roam and handover between both networks. Figure 4-1 shows the integration of mobile networks and WLAN by means of UMA.

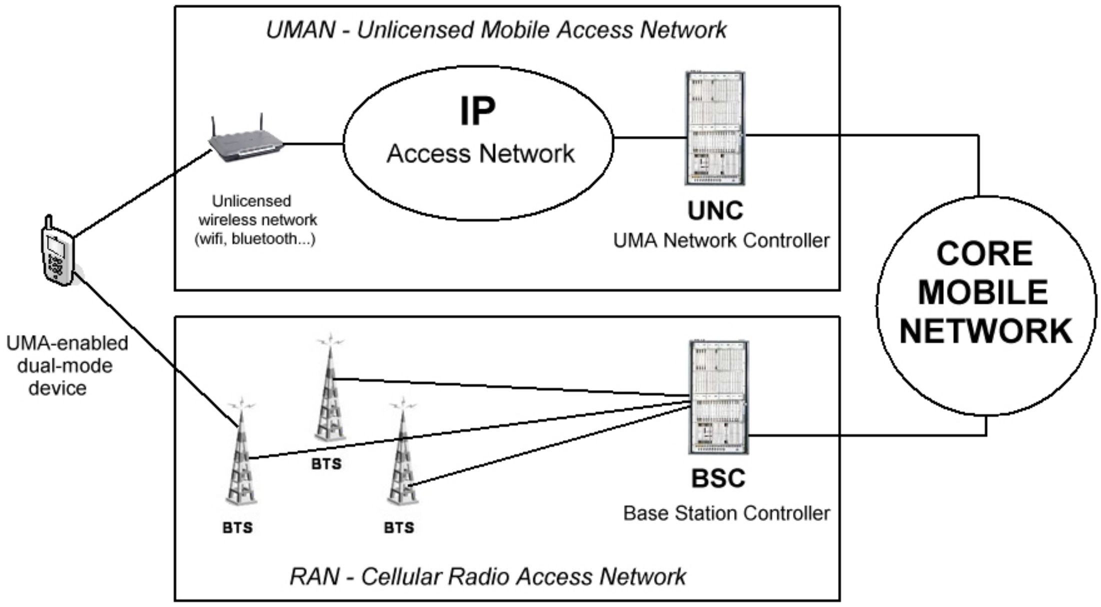

Figure 4-1: Integration of Cellular Access Network and UMA Network.

UMA defines a parallel radio access network, known as UMAN – UMA Network – that interacts with the mobile core network using the current mobile network standard interfaces. In order to connect the UMA network and the mobile core network a new entity is created: the UNC – UMA Network Controller – which interfaces the UMAN to the core Network operating equivalently to a current BSC. Existing business support and operation support systems can support UMAN without any change, thus UMA is integrated directly and smoothly within the mobile network.

For a seamless integration with existing mobile networks and unlicensed spectrum networks it is also needed to define a UMA-enabled handset with dual-mode operation capable to operate within both networks.

4.1.1.1 UMA Operation

The UMA operation procedure is briefly summarized in the next steps:

- A mobile user with a UMA-enabled dual-mode device moves within range of a WLAN to which the handset is allowed to connect.
- Over the IP access network the handset contacts the UNC to be authenticated and authorized to access the mobile network through the WLAN.
- The user's current location information stored in the core network is updated, and all traffic coming from the mobile network is routed to the handset through the UMAN rather than the mobile network.
- ROAMING: When the user moves outside the coverage area of the WLAN to which he is connected, the UNC and handset facilitate roaming back to the mobile network. This process is completely transparent to the user.
- HANDOVER: If the user is on an active call or data session when they come within range (or out of range) of a WLAN, it can automatically handover between access networks with no discernable service interruption. Handover is completely transparent to the user.

4.1.1.2 UMA Architecture

The UMA architecture is based on the following premises:

- Independency of underlying unlicensed spectrum technology (WiFi, Bluetooth…)
- Transparency to existing network infrastructure (e.g. access points, routers and modems).
- Seamless roaming and handover between networks.
- Same identity on cellular networks and WLAN
- Preservation of investment in existing/future mobile core network infrastructure
- No impact on Cellular RAN operations

Figure 4-2 presents the UMA functional architecture.

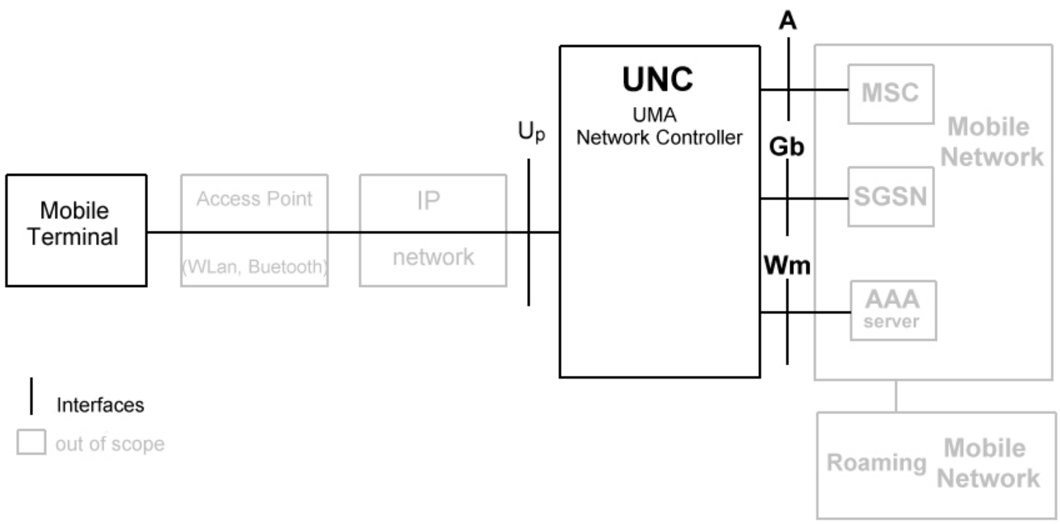

Figure 4-2: UMA Functional Architecture.

UMA specification defines entities shown in black in Figure 4-2 , while grey coloured entities are out of the scope of UMA specification. As shown, only the mobile station and the UNC are in the scope of the UMA specification, keeping the other entities unchanged. This allows the new UMA compliant systems to be transparently integrated within existing networks, both WLAN and cellular networks.

The UNC connects to a unique MSC and SGSN through the A-interface and Gb-interface respectively, providing functions equivalent to a BSC. It connects to the AP through an IP network using the **Up-interface**. Maintains end-to-end communication with the MS, and provides a transparent transfer of L3 messages between the MS and the core network.

In order to obtain access to cellular and unlicensed spectrum networks the mobile terminal should be a dual-mode handset, supporting cellular technologies (GSM, GPRS, UMTS) and unlicensed spectrum access technologies (WLAN, Bluetooth, etc.). The mobile terminal supports an IP interface to the unlicensed spectrum access point, establishing an IP connection that extends all the way to the UNC.

The Up-interface is the interface between the mobile terminal and the UNC and Core Network. This interface operates over an IP transport network and relays GSM/GPRS signalling between the mobile core network and the mobile terminal. The following figure illustrates the Up-interface protocol architecture for the GPRS user plane.

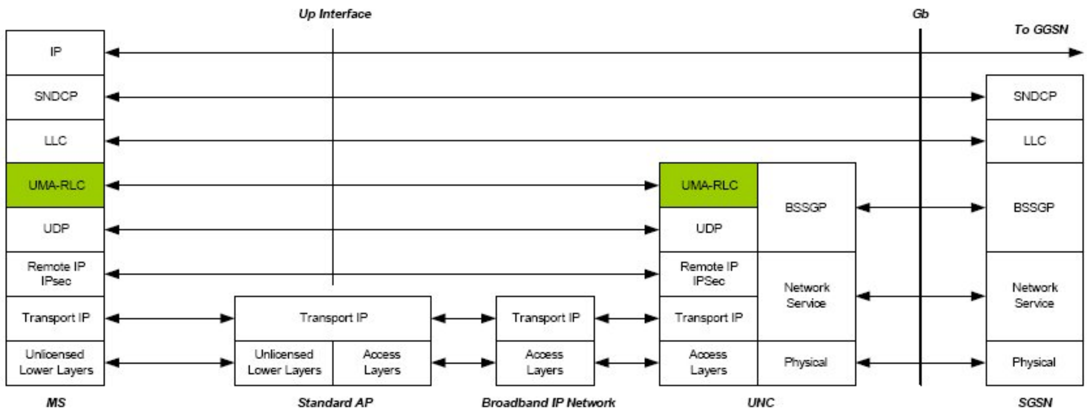

Figure 4-3: Up GPRS User Plane Protocol Architecture.

As can be seen, from the core network point of view, higher layer protocols are carried transparently between the mobile terminal and the SGSN. This allows the mobile terminal to obtain all GPRS services same way as if it was in a cellular base station. The UNC, operating like a BSC, terminates the UMA-RLC protocol and inter-works it to the Gb-interface using BSSGP.

4.1.1.3 UMA and Cellular Network Operators

UMA provides a scenario with many advantages for the integration of unlicensed spectrum technologies with cellular networks. It enables operators to turn existing WLAN networks into seamless extensions of their cellular networks, as the location of the user does not affect the cellular services.

Operators could provide access to their networks by deploying their own UNC infrastructure increasing the coverage and availability of the network. In a second scenario cellular operators will just enable the connection to networks of WLAN operators who will provide WLAN access to mobile users. Another scenario could be considered where third party operators will deploy UNC infrastructure providing interconnection between WLAN operators and cellular networks.

By deploying unlicensed spectrum networks mobile operators would take advantage of the lower cost structure of IP access networks, facilitating the deployment of high-bandwidth IMS services, such as streaming video, video conferencing, music downloads, etc. Even more, an extra income could be retrieved with a minimal cost when allowing the connection of third party WLAN networks, via UMA, to the core network, charging the WLAN provider for the interconnection services.

4.1.2 IEEE 802.21-based media independent handover

The IEEE working group 802.21 (WG 802.21) standardizes approaches to enable Media Independent Handover (MIH). The work focuses on wired and wireless solutions as specified in the IEEE 802-family of standards. Additionally, handovers to cellular systems are evaluated.

The foreseen system consists mainly of three services that are offered and implemented by the Media Independent Handover (MIH) function:

1. Media Independent Event Service (MIES)
2. Media Independent Command Service (MICS)
3. Media Independent Information Service (MIIS)

The MIES is the core functionality required to enable MIH and specifies several technology-independent link layer triggers such as *Link Down* or *Link Going Down* that have to be supported by every standard-compliant MIH function. Applications, such as the MIH function itself, can register for those triggers and will be alerted if the predefined conditions occur. The details of respective trigger implementations, which will be defined separately for each technology, are still under discussion and not described further in the first standard draft from July 2005 [2]. One possible way forward is that the 802.21 WG limits its scope to abstracted service access points (SAPs) and leaves the detailed specification of those interfaces to the standardization boards working on the respective technologies, e.g., IEEE 802.11, IEEE 802.16, 3GPP, or 3GPP2. In addition to the local information retrieval, remote access to SAPs is foreseen. Media independent signalling will be used to access similar interfaces on remote communication peers and enable more centralized management of the whole MIH functionality.

The MICS allows the MIH function to ask the possibly different link layers to perform specific measurements or simply provide the latest measurement results. As the supported actions are predefined and do not offer flexible configuration interfaces the MICS is limited and clearly focused on handover applications.

The MIIS is the third proposed service in the 802.21 draft standard. The MIH function uses the MIIS to gather information about the surroundings and the available networks. The capabilities of these networks are described using an RDF scheme, which is based on XML.

4.2 OMA

The Open Mobile Alliance (OMA) – see www.openmobilealliance.com - has defined a Device Management (DM) framework that uses management objects defined in an XML tree syntax (see reference [5] and [6]) to allow devices to be remotely configured in a generic manner. This uses a SyncML based protocol between client and server entities, which can be supported over many different transport methods. The current scope of this work addresses the need for operators to configure the capabilities of their devices, such as communication parameter settings (for example security attributes), proxy gateways and network settings (DHCP, DNS etc.). The framework relies on each device containing a DM client entity. This entity is currently a device specific software implementation that uses proprietary APIs to access device information and configure the device. Clearly it is advantageous if the DM client implementation could use the ULLA API to permit operation of the same DM client software on different devices. Despite the ULLA API only exposing link layer information and parameters it could still be very attractive, especially as many of the higher layer parameters (such as proxy gateway settings etc.) are traditionally stored within the operating system.

Currently, the OMA DM enablers are being extended to include enablers, such as firmware upgrade, software download and also for DM scheduling. These enhancements allow the DM framework to control the upgrading and deployment of software to devices as well as scheduling automatic configuration operations in response to events (or triggers) or at certain times according to a predefined schedule. These new enablers can also exploit the ULLA API to permit the retrieval of information regarding links available for software download and notification of available networks or changes in link performance related information.

4.3 The Internet Protocol Multimedia Subsystem (IMS)

Modern mobile terminals are more than just mobile phones for voice communication. These devices usually also embed more advanced capabilities like large high definition displays, cameras and resources for more demanding applications. They also have the possibility for data communication based on the Internet Protocol (IP). This gives new possibilities for new peer to peer connection oriented applications.

In order to set up a connection IP based applications must be able to reach the other party. With circuit switched networks this is done by dialling the phone number and setting up the connection. With IP networks this would only work in single provider networks. The Internet Protocol (IP) Multimedia Subsystem also known as IMS is a global system that enables applications on mobile devices to establish peer to peer connections. It is an architecture based on the specification of the Session Initiation Protocol (SIP). The SIP protocol is only a part of it; it is using a wide range of protocols to allow real time services on top of the Universal Mobile Telecommunication System (UMTS) packet –switched network.

The IMS is also designed to be link layer independent; IMS services can be provided over any IP connectivity networks like GPRS, WLAN, xDSL etc.

5. ULLA at Operator levels

The sections describe existing standards in development where the ULLA could be used to simplify the implementation. Some of them also describe new ideas that could be developed based on the availability of the ULLA.

5.1 *ULLA for Device Management*

There are two main ways in which ULLA can benefit the Open Mobile Alliance (OMA) Device Management (DM) framework implementation. The first is in support of the scheduling enabler client[1] and the second is in support of the OMA-DM client parameter configuration functionality (as shown in Figure 5-1). Further information regarding the details of OMA-DM standardized management objects and device management scheduling enabler can be found in references [5] and [6] respectively.

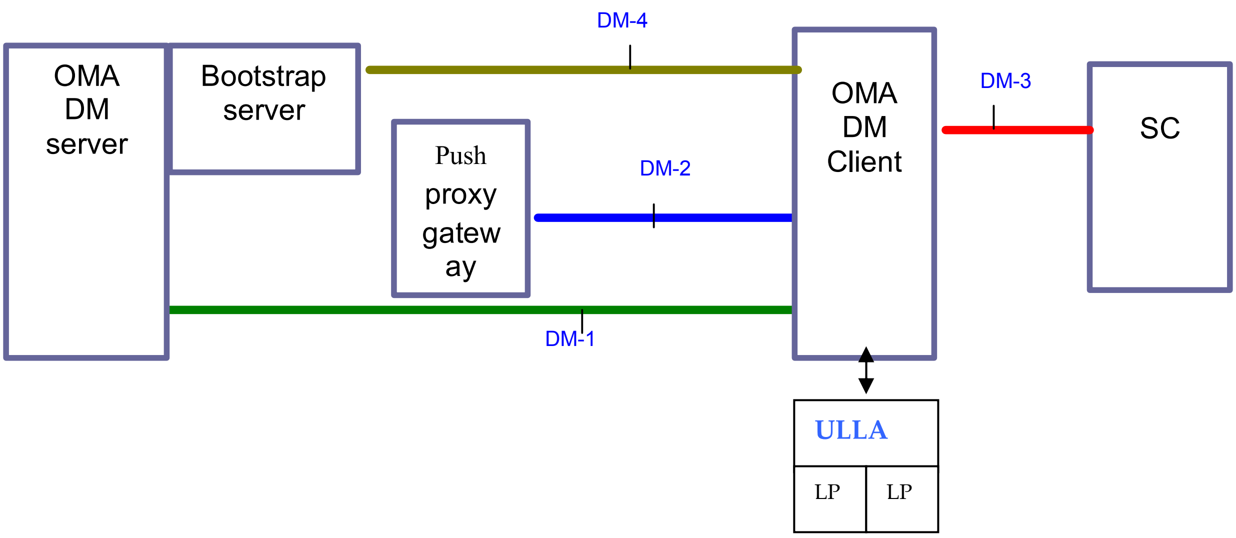

Figure 5-1: OMA DM Architecture with Possible use of ULLA.

The way in which ULLA can provide benefit is in the fact that the ULLA can provide event triggers (notifications) that can be used by the DM client to schedule when to perform configuration changes. The scheduling enabler is deemed to be applicable in a series of use-cases which can clearly exploit the ULLA API, namely:

- Proactive reporting of information to server (using ULLA event based notification)
- On-device tool invocation (for fault or performance diagnosis using link performance information)
- Deferment of firmware upgrades to suitable time (based on network availability using ULLA notifications)

---

[1] The DM Scheduling Enabler Client is the abstract software component in a Device implementation that servers as an end-point of the DM Scheduling Protocol. The DM Scheduling Enabler Client is responsible for receiving, checking, configuring, and running the schedules in the Device. The DM Scheduling Enabler Client is also able to interface with User Interface, OMA-DM Enabler, and other DM Enablers for that purpose.

- Offline operation (self diagnosis displaying event notification messages and information to user to permit configuration of links via ULLA)
- Real-time reconfiguration (reacting to changing circumstances based on ULLA event based notifications)

The ULLA API can enable support for the event notifications and information required and parameter configuration to perform many of the above operations related to the use-cases. ULLA can also enable the setting of parameters by the OMA DM client in a device independent manner. For instance, the WEP or WPA security parameters can be set using the generic *ullaSetAttribute* operation as shown below.

```
AttrDescr_t attrib;
attrib.id = 1;
attrib.linkClass = "ullalinksecurity";
attrib.attribute = "encSecret";
attrib.qualifier = ULLA_QUAL_HARDCODED;
attrib.type = TEXT_DATATYPE;
attrib.length = 10;
attrib.numValues = 1;
attrib.data = « WEP key »;
ullaSetAttribute(attrib);
```

5.2 *ULLA for handover*

Offering seamless handovers is one of the most important requirements for truly mobile systems because communication range is limited. The homogeneous case of intra-technology handovers is solved and working properly in many real-world networks, such as cellular systems. Also the inter-technology case for a small predefined set of networks such as the interworking of GSM/GPRS and UMTS in cellular networks was solved and today operators offer integrated services that automatically choose and control which network to use.

However, two major trends for the handover management emerge in the mobile world: Multi-parametrical handover evaluation and vertical handovers in heterogeneous systems. Whereas two main standardization activities work on the latter problem the former one is more an implementation issue for mobile device or network node developers. The use of ULLA in UMA is described in more detail in section 5.2.2 the combination of IEEE 802.21 and ULLA is explained in section 5.2.1. Before, we shortly comment on how multi-parametrical handover evaluation could be implemented using ULLA.

The traditional handover algorithms use the received signal strength as the main trigger for handover decisions. Other performance metrics could also be monitored to enhance the handover decision process. Examples for such attributes are listed below:

- Error rates measured in the down link based on different units such as bits, blocks, frames or packets
- Number of retransmissions and respective ratios measured in the uplink
- Level of interference and possible identification of interfering technologies
- MAC congestion level

Today's proprietary implementations already use several of those metrics not only for handover management but also for resource optimization. ULLA can be used to generalize this process by using platform- and where possible also technology-independent metrics when evaluating possible handovers. The flexibility of the ULLA and the applied class hierarchy allow very general but also hand-tailored and optimized solutions for single platforms. Link users can query for different metrics but can also combine several attributes in the WHERE-condition and use the resulting query to register for an event-based notification. Such a notification could trigger the analysis and possibly the initiation of a handover. This way, ULLA easily enables multi-parametrical handover evaluations that could also be vertical handovers as described below.

5.2.1 ULLA for IEEE 802.21-based media independent handover

The IEEE working group 802.21 (WG 802.21) is currently standardizing approaches to realize Media Independent Handover (MIH) as introduced in section 4.1.2. One major problem that is stated in an updated version of the initial Project Authorization Request (PAR) for the IEEE WG 802.21 is cited below [1]:

#2 The information necessary to make effective handover decisions is lacking in part because the 802 networks provide insufficient information to the upper layers. Thus there is a need to develop a standard that permits information exchange between mobile terminals and/or networks to enable mobile terminals and/or networks to make more effective handover decisions.

Devices running ULLA and fully functional LLAs for all deployed technologies would solve the former part of this stated problem because ULLA allows gathering link layer information in a well standardized and technology-independent way. The interfaces used between the MIH function and different link layers are not clearly specified, yet. Additionally, those might partially be technology-dependent introducing further complexity. Using ULLA instead would solve these problems and allow media independent analysis. Moreover, all technology-specific details how to measure certain performance metrics accurately would be hidden in the LLA-implementations.

The MIES could easily be implemented using notifications such as those supported by the ULLA Query Processing (UQP). The 802.21 draft standard also foresees that higher layer entities can register for these events as LUs do it for notifications when using ULLA. Again, the ULLA approach is more flexible as each LU can define its own specific conditions for notifications instead of being limited to a predefined subset. However, each LP can be extended to offer the exact 802.21-functionality inside the ULLA framework. For this purpose we introduce a new class in the ULLA class hierarchy supporting IEEE 802.21-based MIH.

The new `ieee80221Link` class includes several variables using the type ULLA_STRING that provide requests how the generic 802.21 events should be implemented for the given LP. The LU would at first query for the string-variables and later on use these strings as conditions when registering for notifications.

The final set of standardized events is not decided, yet, but in the following we list some examples for events that are described in the draft standard [2]:

- Link Up
- Link Down
- Link Going Down
- Link Parameters Change
- Link Handoff Complete

The last example event *Link Handoff Complete* includes higher layer information and thus would have to be implemented by the MIH function running on the mobile device. The former four events can easily be included in the new ieee80221Link-class using attributes such as `ieee80221Link::linkUpClass` or `ieee80221::linkGoingDownCondition`. As IEEE 802.21 events will probably need rather complex queries we implement those as joined queries and include conditions as well as required classes as string-attributes to the newly introduced class.

In the following a partially simplified example for the usage of the newly introduced attributes is given:

```
/*
        Example code for usage of ieee80221Link-class.
        The example LU wants to register for and use the Link Going Down event:
*/

/*
        Function to handle the Link Going Down event.
        Assumed notification-query:
        SELECT ullaLink.linkId, ullaLink.rxSignalStrength, ullaLink.rxQuality FROM
              ullaLink, ieee80211Link WHERE ullaLink.linkId=5 AND
              ieee80211Link.txRetryRatio>0.4
*/
void handleLinkGoingDown (RnId_t rnId, UllaResult_t res, void* privdata) {
        int linkId;
        int rxSignalStrength;
        int rxQuality;

        ullaResultIntValue(res,1,&linkId);
        ullaResultIntValue(res,2,&rxSignalStrength);
        ullaResultIntValue(res,3,&rxQuality);
        ullaResultFree(res);

        /* Perform any required action. */
}

LuDescr_t myDescriptor;
LuId_t myId;
myDescriptor.name = "GOLLUM Demo Link User";
myDescriptor.Description = "A 802.21 demo LU";
myDescriptor.Version = "1.0"
int error;

/* Register Link User */
error=registerLu(&myDescriptor,&myId);

...

UllaResult_t myResult;
#define LENGTH 64
char[LENGTH] query, condition, class;
int linkId = 5; // we assume we want to monitor linkId=5

/*
        Retrieving string specifying technology-specific condition for Link Going Down
        event
*/
sprintf(query, "SELECT ieee80221Link.linkGoingDownCondition FROM ullaLink,
        ieee80221Link WHERE ullaLink.linkId=%d;", linkId);
error=requestInfo (luId, query, &myResult);
ullaResultStringValue (myResult, 1, &condition, LENGTH)
ullaResultFree (myResult);
/*
        The resulting condition could be:
        ieee80211Link.txRetryRatio>0.4
*/

/*
        Retrieving string specifying technology-specific classes required for Link
        Going Down event
*/
sprintf(query, "SELECT ieee80221Link.linkGoingDownClass FROM ullaLink, ieee80221Link
        WHERE ullaLink.linkId=%d;", linkId);
error=requestInfo (myId, query, &myResult);
ullaResultStringValue (myResult, 1, &class, LENGTH)
ullaResultFree (myResult);
/*
```

```
        The resulting class(es) could be:
        ieee80211Link
        The string will be empty if only attributes included in ullaLink are used.
*/

if (strcmp(class,"")!=0) then
        sprintf(query, "SELECT ullaLink.linkId, ullaLink.rxSignalStrength,
                ullaLink.rxQuality FROM ullaLink, %s WHERE ullaLink.linkId=%d AND %s;",
                class, linkId, condition);
else
        sprintf(query, "SELECT linkId, rxSignalStrength, rxQuality FROM ullaLink WHERE
                linkId=%d AND %s;", linkId, condition);

/*
        The resulting query could be:
        SELECT ullaLink.linkId, ullaLink.rxSignalStrength, ullaLink.rxQuality FROM
             ullaLink, ieee80211Link WHERE ullaLink.linkId=5 AND
             ieee80211Link.txRetryRatio>0.4
*/

RnDescr_t myNotification;
myNotification.count = 0; // valid until explicitly cancelled
myNotification.period = 0; // event-based
myNotification.handler = &handleLinkGoingDown;
myNotification.privdata = NULL;
int rnId;

error=requestNotification (myId, rnId, query, myNotification);

...

/* Cancel notification
error=cancelNotification (myId, rnId);

/* Unregister Link User when closing application */
error=unregisterLu(myId);
```

Figure 5-2: Code example for the ieee80221Link class.

The returned attribute `ieee80221Link::linkGoingDownCondition` could, e.g., have the value `ieee80211Link.txRetryRatio>0.4` and the involved classes given by `ieee80221Link::linkGoingDownClass` are `ullaLink` and `ieee80211Link`. Since `ullaLink` is always included we simply add `ieee80211Link`. The example further assumes that we want to monitor a WLAN-link with linkId=5.

If one device manufacturer has developed more advanced methods to control the event process these can be included in the implementation of the `ieee80221Link`-class. The introduced query-strings can use joined queries and incorporate information coming from manufacturer-specific additional classes for very specific attributes. This way the interface towards the higher layer MIH function looks always the same but the technology-specific implementation in the LLA is completely up to the LLA-programmer.

The default value of the introduced condition-attributes should be defined by any involved standardization body for basic compatibility-reasons. It can still be technology-specific as, for example, cellular systems usually work with block error rates (BLER) but WLANs work on frame-level and thus would include frame error rates (FER). Also the used SNR-thresholds will probably be technology-specific as the receiver-sensitivities will be different.

Besides the MIES the draft standard also describes the Media Independent Command Service (MICS). Although the name indicates similarities to the ULLA Command Processing (UCP) the MICS is not as powerful as the respective UCP features. The MICS is mostly designed to enable the MIH functions to request lower layers to perform certain measurements or simply provide latest measurement results. Such functionality is covered by the UQP and simple configuration updates can be realized using the ULLA-command *setAttribute*.

The MIH function uses the third 802.21-service, the MIIS, to gather information about available networks and their capabilities, which are described using an XML-based RDF scheme. ULLA could easily be extended to support also the MIIS. The description of the available networks and their capabilities is already supported by the ULLA framework as it is defined now. The additional support for the RDF-syntax can be added to the `ieee80221Link`-class introduced above. New attributes in this class could provide all the needed information using the syntax and semantic finally standardized by the IEEE WG.

ULLA is very appropriate for implementing 802.21-based MIH. All required features are supported by the latest ULLA or could easily be added when the final 802.21-standard is available. The proposed new class `ieee80221Link` would include all needed extensions but cannot be defined before the final standard is ratified. Additionally, ULLA is able to further abstract the link layer access and enable technology-independent handover-management. The latest standard draft still relies on technology-specific interfaces that are to be defined by the respective technology-standardization. ULLA would solve this problem and offer the same interface for all deployed technologies. Moreover the implementation of the 802.21-events is very flexible and new developments can be included without any need to change higher layer entities. End-user devices as well as network entities that run ULLA could ease the adoption of MIH in today's operator networks.

5.2.2 ULLA for UMA

Dual-mode devices are becoming more popular these days providing access to cellular networks and WiFi networks. But the current status neither satisfies users nor operators because of the lack of integration and interaction of both modes, cellular and WLAN. Many operators are deploying wireless networks all around in order to provide an extra service to their subscribers but just offering an Internet access service whose billing is integrated with the main subscription. UMA presents a unique opportunity for operators to closely integrate both access technologies. By deploying UNCs which will connect the already existing WLAN access points with the mobile core network, the services and capabilities offered to the subscribers will be amazingly improved.

Nevertheless, to obtain a fully functional UMA network not only UMA network infrastructure has to be deployed, but UMA-enabled mobile terminals need to be available for subscribers. A UMA network will be useless if users do not have access to the adequate devices. Further, a dual-mode terminal which would be UMA-enabled but would have restrictions in current WLAN operation, because of being UMA oriented, will not satisfy user expected experience, delaying it success.

ULLA appears to be the proper solution to integrate both approaches, UMA and IP wireless networks. As UMA enables the connection to unlicensed spectrum networks via 802.11 or Bluetooth (PAN profile is required), UMA could take advantage of ULLA capabilities to manage the available links and to select the most adequate technology. Therefore ULLA could be integrated within the UMA mobile terminal architecture to manage the access to the unlicensed spectrum links, facilitating the handovers between technologies.

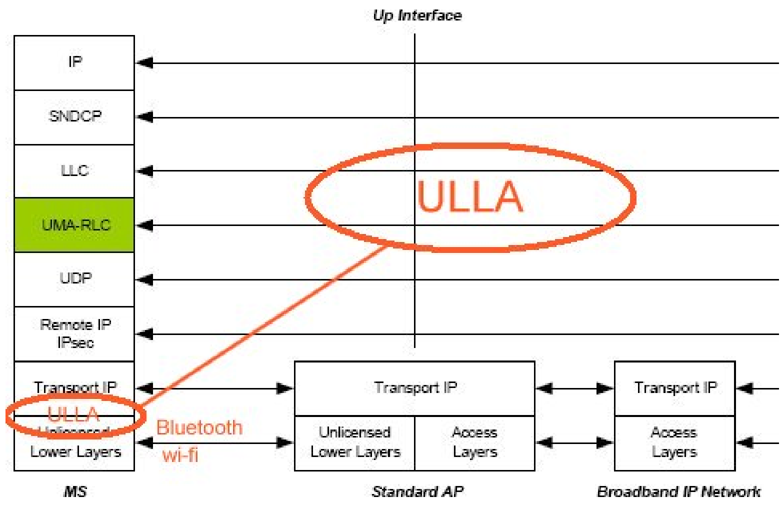

Figure 5-3: Integration of ULLA in an UMA-enabled mobile terminal.

The UMA specification provides the following recommendation for the 802.11 mobile terminal capabilities:

- *Scanning: When the MS is in GERAN/UTRAN mode, the MS should periodically scan for 802.11 coverage. The interval for scan is implementation specific, depending on power conservation strategies. For example, the MS could wake up the 802.11 radio every 2-3 seconds and conduct an active scan (lasting less than 100 ms).*
- *Joining: The MS should use SSID, security settings and RSSI, among other parameters, to select which AP to join.*
- *Authentication: The MS should support WEP and WPA with PSK. It could optionally support 802.1X authentication (and one or more upper layer authentication protocols e.g. EAP/TLS) for large enterprise environments.*
- *Encryption: The MS should support WEP (RC4) and WPA (TKIP) encryption. It could optionally support WPA2/802.11i (AES) encryption.*
- *RSSI: The MS should be able to (internally) measure RSSI values between -45 dBm and -86 dBm with a step size of 1dB. Inter-AP "roaming threshold" should be set such that there is reasonable opportunity to discover other APs when the signal from the current AP drops (before the MS is forced to switch to GSM GERAN/UTRAN mode).*
- *Roaming: The MS 802.11 driver must be able to keep upper layers unaffected by change of AP (provided the IP address does not change). Communication interruption due to change of AP should be kept below 100 ms (provided the re-authentication latency is below this number). In particular, a DHCP lease renewal should not be required before communication resumes on the new AP (i.e. assume IP address does not change, but then verify this is true, in order to recover if indeed the IP address has changed).*
- *Delay: The MS should implement adaptive jitter buffer to keep this delay component low. The MS should internally prioritize voice packets ahead of any other data packets that are to be transmitted. The MS should support 802.11e EDCA (or WMM) to allow prioritized access for voice packets to the medium. If the AP does not support 802.11e EDCA (or WMM), the MS could "simulate" this behaviour by using a smaller contention window and inter-frame spacing for voice packets, relative to other data packets. In order to do so, the voice packets*

must be marked for preferential treatment when being delivered to the 802.11 driver from the application.

- *Loss: The MS should use intelligent packet loss concealment algorithm to mitigate packet loss.*
- *Network QoS: The UMA-MS should check DSCP/ToS information in incoming IP packets from the UNC and if different from what currently used, copy received DSCP/ToS and use it for outgoing IP packets.*
- *Power save/sleep mode: The MS should use power-save/sleep mode as follows: When in GERAN/UTRAN mode, the MS should periodically scan for 802.11 coverage as specified above. When in UMA mode but not on an active call, the MS should use power save mode with an implementation specific sleep interval e.g. waking up 1-2 seconds to check for incoming traffic, maintain synchronization and make RSSI measurements. When in UMA mode and on an active call, the MS may or may not use power save mode.*

Despite the fact that many of the previous features are already supported by ULLA and ULLA classes, such as *ieee80211Link*, the access to the wireless networks within UMA managed by ULLA would probably require a new link class, *umaLink,* in order to support the special needs of the wireless link classes for the UMA architecture. Some other of the desirable features shall be implemented by means of the *Link Manager* or the *Connection Manager* entities, such as AP roaming, scanning policies, power mode selection, etc.

Some additional attributes that would be required are:

- IMSI*, obtained from the SIM and requested by the UNC during the registration process. It is used by the UNC to index the appropriate MS record when the UNC receives a BSSMAP PAGING message.*
- *UMA Cell_Id,* which can be used by the core network for location or charging purposes.
- *UNC_Id*, Identifier of the UNC to which the terminal is connected with.

The complete definition of this class is out the scope of the present document, and the previous attributes are shown as an example. A deeper research of the UMA specification is required.

5.2.2.1 Integrating UMA and ULLA: Operator Role

UMA provides a scenario with many advantages for the integration of unlicensed spectrum technologies with cellular networks. It enables operators to turn existing WLAN networks into seamless extensions of their cellular networks, as the location of the user does not affect the offered services.

In order to facilitate and accelerate the delivery of UMA-enabled terminals to the users, operators play a major role, with two main activities

1. Deployment of UMA infrastructure
2. Support of UMA terminal manufacturers

5.2.2.2 Deployment of UMA Infrastructure

The deployment of UMA Infrastructure consists of the deployment of a complete and fully functional infrastructure, which will allow users the same experience when connecting to the wireless network through an unlicensed spectrum network as in the case when using the cellular access network.

To accomplish these expectations, operators should work on the following guidelines:

- *Deployment of operator-owned UNC directly connected to the operator core network.*
- *Connection of current operator-owned internet AP access to the UNC in order to complete the UMA network.*
- *Sign agreements with wireless internet service providers allowing access to the operator's UNC of traffic coming from their AP. This will rapidly increase the coverage of the UMA access service, and improve the operators' coverage, as every WLAN owner would like to increase the value of their network with the added value of the UMA access.*
- *Roaming agreements with third party UMA providers. Some companies, in particular owners of public WLANs, would like to provide their own UNC infrastructure, but they will lack the connection to the cellular core network. Cellular operators should facilitate the interconnection of this UMAN to their core network.*

5.2.2.3 Support of UMA mobile devices

Although operators are not responsible for the design of mobile devices, they have a big influence on the decision process of manufacturers. If operators are interested in a technology to be included in the terminals, they can encourage mobile terminal manufacturers to adopt the new technology by subsidizing terminal acquisition to the users, and providing new exciting services that would demand the need of new terminals.

The adoption of ULLA would facilitate the access to both cellular and unlicensed spectrum networks, would provide the user with a powerful device that increases the options for the user to be permanently connected through almost all the available networks, and giving and alternative access for broadband connections. Finally, because ULLA has been designed to be easily extensible, the adoption of the ULLA would be a guarantee to work also with future technologies.

Operators could provide their own ULLA Link Manager or Connection Manager with specific policies defined for their network access. The manager would connect the user to the best available network taking into account the requirements the user is demanding and the networks' availability. As an example we could mention a manager that would contain the needed information to access the networks to which the operators have signed agreements, liberating the user for the tedious task of connection searching.

At the end, the seamless connectivity that simplifies the user experience will increase the demand for data services, enabling the development of new services and technologies such as IMS.

5.3 ULLA for Network Management

In forthcoming wireless communications scenarios, mobile terminals are likely to support several radio technologies. The number of different RATs (Radio Access Technologies) used by a terminal could be considerably large, in particular when those ones are used in a global scale. In order to cope with the increasing heterogeneity of the available networks, the technologies need to interact between them. This adds complexity to the terminal, as it needs to monitor the radio environment of all the subjacent RATs.

On the other hand, at an operator level, both the management functions and the network infrastructure that may be used when heterogeneous radio access technologies coexist within the same network should be identified. Network management systems need to be flexible enough in order to adapt their resources and the interworking levels among different network elements.

3GPP has defined the different procedures for the network management in the UTRAN Network [7], [8], [9]. Radio network management has been specified in terms of performance measurements and QoS monitoring process. These procedures are executed at Network Element level, where the operator could use the ULLA in order to obtain this information.

Due to the heterogeneous coexistence of multiple radio technologies within a same network element (BS and RNC), the traffic distribution could suffer from a high variance. In these scenarios, the reconfiguration of the network could be triggered by the awareness of some network parameters. Network elements could use the ULLA in order to carry out performance measurements and QoS monitoring. A specific link user, within the network element could request notifications for certain QoS parameters, and once reached a QoS threshold, the network element could send an alarm to the network upper management layers. Because the ULLA is technology agnostic and can be used with current as well as upcoming type of links, the development of network management solutions based on the ULLA would guarantee that the software developed now does not have to be replaced when new wireless technologies are rolled out.

5.4 ULLA for Network Evolution

Wireless access networks are constantly evolving and new technologies are introduced in order to enable additional services. In this section we comment on the usage of ULLA with some examples for such evolutionary network extensions.

5.4.1 ULLA and IMS

On public Internet there is no guaranty for the quality of the connection; packets can get lost or arrive out of order, delays can be high. With IMS (Internet Protocol Multimedia Subsystem) access and transport layer will work together to provide end to end quality of service. Via IMS the terminal will negotiate its capabilities and its QoS requirements during the SIP setup. When the parameters are negotiated at application level the terminal will reserve the resources from the access network. It is assumed that operators will negotiate service level agreements for guaranteeing the required QoS in the interconnection networks.

IMS will not only support QoS it will also provide at least a similar level of security as the corresponding circuit switched and GPRS network. IMS will also handle authentication before services can be accessed.

Charging is also part of IMS. It will support online and offline charging. It will also be able to charge per media component like audio or video.

IMS uses a User Database that will be used for handling the calls/sessions. It contains the subscription-related information (user profiles), performs authentication and authorization of the user, and can provide information about the physical location of users. Special Application and Media servers will also allow third party providers an easy integration and deployment of their value added services to the IMS infrastructure.

As IMS supports different access networks ULLA can not only be used on the terminal for enabling easier integration of the different link layers it can also be used in the application and media servers for better QoS and charging support.

5.4.2 Future Networks

Although ULLA seems most likely to be found on mobile terminals there are also possibilities to use ULLA inside network core components.

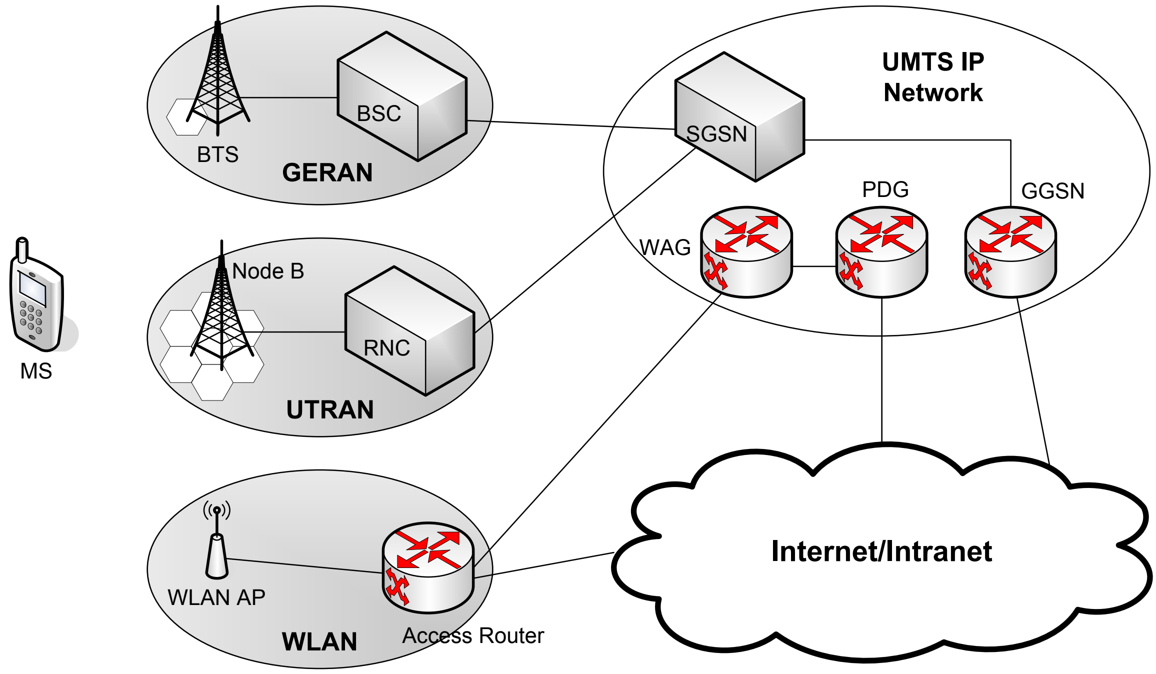

Figure 5-4: Internetworking WLAN and PLMN.

Figure 5-4 depicts a configuration of a Public Land Mobile Network (PLMN) with WLAN interworking. Typically a mobile station (MS) will connect to a base station. Depending on the network it will connect to a Base Transceiver Station (BTS) for GPRS or a Node B for UMTS. A BTS is a network component which serves one cell. The BTS is connected to a Base Station Controller (BSC). This is a network component in the PLMN with the functions for control of one or more BTS. A Node B is a logical network component which serves one or more cells. A Radio Network Controller (RNC) is a network component in the PLMN with the functions for control of one or more Node B.

The first possible usage of ULLA can be found in a Node B. This network component controls multiple cells. ULLA can be used to simplify the monitoring of the links corresponding to the different cells. Hence the ULLA could be used to provide a uniform API to control the handovers between the cells that the Node B is controlling. The result of a better management of the links could be a better load balancing of the traffic in the different cells as well as a better perceived quality of the service. The advantage of using ULLA would be that the algorithms developed for this technology would be usable for future technologies, too.

3GPP is working on WLAN integration. It will support WLAN as access network independent if operated either by a 3GPP operator or a third party. The integration levels have been categorized into six hierarchical service "scenarios" [TS 22.934], as listed and further described in section 4.1.

In UMTS Release 6 scenarios 1-3 will be supported. In release 7 even WiMAX integration is foreseen. With this integration support new types of base station might be developed that support multiple access network, GERAN, UTRAN, WLAN and even WiMAX. Using ULLA as core component in these base stations would simplify development a lot. With the extensibility that ULLA offers it would not be too difficult to add new access networks to ULLA enabled base stations.

5.5 ULLA for Resource Optimization

It is now common to see mobile terminals (phones or PDAs) coming with several embedded network technologies. This trend will actually accentuate with the development of new wireless technologies (UWB, Zigbee…). On the other side, these devices are still resource constrained in particular battery life remains a critical resource.

Through the usage of the ULLA, the software running on mobile terminals could become network aware and start applying policies to limit their usage of wireless networks. For example, network awareness would enable applications to choose between different available networks using different criteria such as battery consumption, cost or bitrate.

Once a link has been chosen, the ULLA can also be used for dynamic monitoring of the link characteristics. Through its asynchronous programming model, the ULLA optimizes battery life by allowing applications to monitor network conditions without having to rely on costly polling operations, dynamically monitoring link characteristics can be used to trigger different forms of content adaptation that could lead to further resource usage optimizations.

Finally, because the ULLA is technology agnostic, and can be apply to future technologies, the software developed for these operations can be kept and reused with new types of links.

6. New Enhanced Services

In this section, we present some new ideas for possible new and enhanced services that could be provided by operators based on the usage of the ULLA.

6.1 ULLA for Voice

ULLA can help improving the quality of voice calls in an Unlicensed Mobile Access (UMA) architecture. In fact, UMA requires that voice calls use Voice-over-IP (VoIP), therefore a packet-based rather than circuit-switched approach.

ULLA role can be classified as follows:

- Roaming prediction: ULLA can notify the voice application upon changes in detected infrastructure (entering or leaving an access point) so that handover is anticipated
- Voice quality improvement: as the conditions of a WLAN can vary over time (either due to congestion or the wireless link becoming noisy), ULLA can help monitoring the dynamic network state and help the voice application take countermeasures; for example,
    - a different voice codec can be selected or
    - a WLAN access point that operates on a less crowded channel can be used.
    - In case of excessive packet losses, ULLA can notify the voice application to apply error concealment techniques (like Audio Adaptive Playout).
    - Cross-layer optimisation techniques can also be used to fine tune MAC and PHY parameters according to ULLA network condition estimates.

6.2 ULLA Unique ID and Certification Authority

Every access network has a Unique Identifier to identify the device/subscriber on the network.

Mobile networks can use the International Mobile Equipment Identifier (IMEI) to identify the terminal or the International Mobile Subscriber Identifier (IMSI) to identify the subscriber. Similar the MAC address (unique address of a networking equipment interface) is used with 802.3 types of networks (including 802.11). The problem is that there is no unique identifier that can be used across several different access networks. ULLA might employ a unique ID (UID) scheme that helps identifying an ULLA enabled device accessing the network on a higher level.

Such an UID may serve as key for access control, accounting and billing purposes and should be enforced by appropriate security measures. For example, a device exhibiting an ULLA UID could be challenged by its peer to prove the validity of its UID, which can be verified by a Certificate Authority (CA).

Having a secure ULLA UID and a CA enforcing this UID would be a great step forward towards mobile communications security. For example, an application on an ULLA enabled device could be used for micro payment services based on the ULLA UID without further proprietary security schemes. Another application for a secure ULLA UID would be automated authentication for machine-to-machine communications in industrial environments.

6.3 *ULLA for Content Adaptation*

The ULLA UID mentioned above could also be used as a key to a Device Capability Database (DCD), if the UID contained information on the make and model of a specific ULLA device. The Device Capability Database could be consulted for specific purposes such as delivering multimedia content to an ULLA device in a format that is suitable for that device (e.g. dependent on colour depth, display size and available codecs).

Not only the device capabilities but also the used link layer can be taken into account. When using a network with a higher available bandwidth, better quality can be used to encode the multimedia content (e.g., higher source coding rates allowing higher resolutions, frame rates, etc.).

The Device Capability Database can be implemented either in a stand-alone mode (one DCD per network where all ULLA devices used in that network are registered), or in a master-slave mode (hierarchical model, where a few master nodes synchronize their information with many slave databases in several networks). Device manufacturers would publish a device capability profile to that database in formats already used today, e.g. RDF and OMA UAProf. Those formats and schemes may need some extensions to cater for additional, ULLA-related information.

7. Conclusion

With the progress of wireless networks and the threat of cheap Voice communications over the internet, operators rush to integrate, in traditional cellular networks, other types of wireless technologies that will enhance the ability for users to be connected to the Internet. Of course this integration should be made in such a way that users do not see any differences other than a better perceived service. In particular a great deal of attention is given to solutions that hide, from users, the usage of different network technologies (transparent vertical handover).

After presenting an overview of the Unified Link Layer API this document has presented this trend as well as the different standardization efforts that are taken place. The GOLLUM consortium thinks that there is there a great opportunity to use the ULLA both on mobile terminals as well as in the Radio Access Network part of the operator networks. For example several standards are in development to provide transparent handover support between Cellular and WiFi networks. To enable smart policies guaranteeing a "make before break" handover solution, the dynamic monitoring of links becomes necessary. By providing asynchronous notifications and the ability to express any type of notifications, ULLA is of primary interest and could actually be used on the phones as well as in the RAN to implement these protocols. Similarly, by providing notifications, the ULLA could be useful for device management. On the network side, ULLA could be used to simplify link characteristics collection and hence network management.

In all these examples, the ULLA could help in the development of algorithms that stay technology independent and hence ready to be used also with incoming new technologies and standards. The GOLLUM consortium hopes that through this document, considering incoming standards that will shape the networks deployed by operators in the years to come, the applicability of the ULLA to operator world has been fully demonstrated.

8. References

[1] IEEE-SA Standards Board, "Project Authorization Request (PAR) 802.21".

[2] IEEE WG 802.21, "D00.01: Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", July 2005.

[3] www.umatechnology.org (visited February 2006)

[4] 3GPP TS 43.318 "Generic access to the A/Gb interface" within the GERAN (GSM/EDGE Radio Access Network) TSG (Technical Specifications Group)

[5] "OMA Device Management Standardized Objects", Candidate Version 1.2 – 07 Jun 2005, Open Mobile Alliance OMA-TS-DM-StdObj-V1_2-20050607-C

[6] "OMA Device Management Scheduling Enabler"; OMA-AD-DM-Scheduling-V1_0-20051018-D

[7] 3GPP 32.101 Telecommunication management; Principles and high-level requirements.

[8] 3GPP 32.104 Telecommunication management; 3G Performance Management

[9] 3GPP 32.800 Telecommunication management; Management Level Procedures and Interaction with UTRAN