OMTP PUBLIC

OMTP

RECOMMENDED PRACTICES FOR CONNECTED APPLICATIONS

VERSION:	Version 1_5
STATUS:	Approved for Publication
DATE OF PUBLICATION:	8th October 2007
OWNER:	OMTP Limited

CONTENTS

1 INTRODUCTION

1.1 DOCUMENT PURPOSE

Many mobile applications being delivered to the market today - such as push email applications, Instant Messaging applications and blogging applications - connect via the Operator's network to send and receive user data. Such Connections often happen automatically either using a push or pull mechanism so as to provide the user with a virtual 'always connected' experience. Many open Terminal platforms available today allow the user to run many Connected Applications simultaneously thus allowing the user to remain connected to many different social networks simultaneously. For example, many users run their email application and Instant Messaging application at the same time so that they remain connected to their work contacts and personal contacts at the same time. It is therefore important that vendors of Connected Applications such as these follow recommended best practices for Connected Application development in order to offer the user the best user experience whilst conserving battery power and network utilisation, both of which, if poorly managed, will lead to user dissatisfaction.

This document contains a set of recommended practices that outline the expected behaviour of a Connected Application as seen from a Terminal and network Operator perspective as well as a set of recommendations for the Connection management on mobile Terminals.

The recommended practices are aimed at influencing the application experience on the Terminal directly.

1.2 BUSINESS RATIONALE

Connected Applications (such as push email) have already proven that they are a compelling mobile service and the market for Connected Applications has plenty of scope to grow.

To enable the next tranche of Connected Applications, user experience barriers and infrastructure limitations need to be addressed. For example, such applications should:

- manage usage of network Connections to facilitate use of several services at the same time, such as email, IM and browsing.
- allow the start of Connected Applications on Terminal power-up and resume Connection as appropriate.
- specifically, for applications such as mobile email, enable viewing of rich attachments such as Microsoft Word, Microsoft Excel, Adobe Flash and video where the Terminal has support for these attachment types.
- ensure optimal utilisation of battery power and network resources.

Since the user's experience of a Connected Application is determined by several parties with ambiguous responsibilities (e.g. in the case of UMTS/GPRS networks, should the network support multiple APNs or should Terminal manufacturers/application vendors re-use one APN?) an industry standard Recommended Practice will help.

The Recommended Practices for Connected Applications task will underpin Operator marketing messages enabling the industry to offer a wide range of Connected Application to their users with confidence.

This will help de-fragment the mobile market by providing consistency in the implementation of Connected Applications on smart-phones and in how the Terminal platform is used. Applications vendors can then differentiate with features rather than implementation details.

1.3 INTENDED AUDIENCE

- Application developers
- OS vendors
- Terminal vendors
- OMTP groups and other OMTP tasks

1.4 CONVENTIONS

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [1].

- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
- MAY: This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the

item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

The requirements within this document are uniquely identified using the following format:

XAPP-#### where:

- XAPP is the 4-letter acronym identifying the subject of this OMTP document
- #### is a number that uniquely identifies the requirement (e.g. 0020) and which is unique within this OMTP document.

2 DOCUMENT SCOPE

2.1 KEY OPERATOR CONSIDERATIONS

This document describes the expected on-Terminal behaviour of a Connected Application. A Connected Application expects network access via cellular networks or via other alternative wireless networks (WLAN, WiMAX, Bluetooth, ZigBee, NFC, etc.) and may have the ability to configure Connections.

The requirements are grouped in the following areas:

- Installation & Configuration
- Connection Management
- Battery Life Management
- User Experience

Some Connected Application desired functionalities have critical inter-dependency with the Operating System and Application Framework supplied on the Terminal. The resulting requirements are covered in the final section:

- Terminal Requirements

Positioning information on Connection Management and Battery Life Management is included in Appendices 1 and 2 of this document.

2.2 APPLICATION SIGNING

Increasingly, Terminals are supporting an Application Security Framework to limit risk of malware (see also OMTP ASF). Where Terminals support such frameworks, it may impact the way applications install and execute. Therefore, to ensure best functionality, third party Connected Applications may need to be signed before they can be installed or used.

Application vendors need to ensure that their Connected Applications are signed using a signing programme aimed at the possible Application Execution Environments on target Terminals.

Application vendors should keep in mind that in order to satisfy some of the recommendations in this document there may be dependencies on the applicable signing programme. For example the Connected Application may be subject to additional signing programme criteria.

2.3 Other Requirements For Successful Network Usage

This document is a necessary part of a complete Connected Applications picture. Successful network usage by Connected Applications often imposes additional requirements on client behaviour, network configuration, and server behaviour. The specifics depend on the application network usage characteristics.

The document ‘Deployment Considerations for lemonade-compliant Mobile Email’ [2] contains requirements and considerations specifically for mobile email and which may be generally applicable to many applications which use session-based TCP.

3 General Requirements

3.1 Installation & Configuration

REQ. ID	Requirement	Rationale
XAPP-0010	The installation or uninstallation of a Connected Application SHALL NOT modify network connectivity settings common to the platform and other Connected Applications.	Terminals using an open OS will allow users to install multiple applications from different application vendors which provide the same, or similar, functionality. For example, a user may install multiple push email clients from different application vendors. Application vendors need to ensure that their applications do not therefore delete or modify common settings that are shared with other applications on the Terminal.
XAPP-0020	Application developers SHOULD provide means to enable users to 'clean' the Terminal such that no traces of the Connected Application are left on the Terminal (e.g. via master reset, via application uninstallation).	This is useful and necessary when user wants to de-install the application because they no longer need to use it or because they want to sell the Terminal
XAPP-0030	A Connected Application preconfigured to auto-start and make a network Connection upon Terminal power-up, SHOULD do so unless otherwise restricted by the Operator or User.	To ensure the optimum user experience.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0040	Application settings SHOULD be provisioned automatically by the Application vendor, e.g. by a provisioning mechanism triggered by the Terminal's IMEI or IMSI (if available to the application to read) or by another mechanism provided by the Application vendor.	Asking a user to manually enter server settings is a barrier to service adoption and utilisation. The ideal solution is for the installation package to either contain, or download, the settings for a particular Operator or enterprise service automatically.
XAPP-0050	For ease of discoverability by the user the Connected Application SHOULD integrate with native Terminal applications (such as an email client integrating with the native messaging function in a Terminal) or alternatively provide a way to enable the user to easily discover the Connected Application on their Terminal	Integrating with native applications or the applications' selection menus is preferred for consistency. However, if the 3rd party Connected Application can be found by the user easily that meets the spirit of the criteria.
XAPP-0060	Connected Applications SHOULD utilise embedded viewer capabilities on the Terminal for displaying videos, Adobe Flash multimedia, Microsoft Word, Microsoft Excel, Microsoft Powerpoint, PDF files, etc. The Connected Application SHOULD use the mechanism supplied by the Terminal to connect with the appropriate viewer.	Many Terminals ship with various document and media viewers preinstalled. Applications should wherever possible make use of these preinstalled viewers because it: 1. saves memory. 2. ensures a more consistent user experience.
XAPP-0070	Connected Applications SHOULD support the capability for automatic over-the-air application upgrade.	Software/network updates may require an updated version of application to be sent to users.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0080	Connected Applications SHOULD provide a mechanism to allow the Operator to force a user to disconnect from the network	Operators may provide users with access to a service on a trial basis and therefore will need a way to logout users when the trial period expires. Operators may only check that a user has subscribed to a service when they login and as such the ability to force a user to logout will in effect facilitate a subscription check when the user tries to log back into the service.
XAPP-0090	Where a Terminal offers the user a backup feature, the Connected Application SHOULD integrate with, and utilise, this same mechanism for backing up user data stored within the Connected Application.	Any user data or settings stored by the Connected Application should be backed up by the user whenever they use the generic backup technology in their Terminal. If the user then loses the data in the Terminal due to hardware failure or theft then they can restore the settings and data for all of their applications using the same mechanism. Terminal-wide backup feature can include over-the-air network backup, or backup to a PC or backup to removable media.
XAPP-0100	Connected Applications SHOULD allow the backup of both settings and data.	To ensure the best user experience

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0110	Connected Applications SHOULD respond to backup and restore requests initiated by the Terminal backup/restore application.	To ensure the best user experience
XAPP-0120	If user intervention is needed to configure the Connected Applications settings the application vendor SHOULD offer a wizard type tool.	To ensure the best user experience.
XAPP-0130	Connected Applications SHOULD provide a configuration option that the network Operator can use to specify the allowed Connections and their priorities without prompting the user.	A network Operator high value service may be tied to a specific Connection (e.g. to a particular APN in a UMTS/GPRS network). If the Connected Application connects to a different Connection it could undermine the business model for the service. Connecting using the wrong Connection may lead to the customer incurring unnecessary charges.

3.2 CONNECTION MANAGEMENT

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0140	The Connected Application SHOULD use a Connection (or a Connection Group if supported) that has been specified by the network Operator if one is available.	Using a specific Connection for a Connected Application enables Operators to charge correctly even when roaming. If there is a choice of Connections, and the Connected Application is provisioned as a Network Operator service, then it should use the Operator's preferred Connection.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0150	When a Connected Application is installed and wishes to create a new Connection (e.g. in the case of a UMTS/GPRS network with a particular APN configuration), it MUST check whether a suitable Connection already exists and, if so use the existing Connection.	To ensure the optimum user experience.
XAPP-0160	The Connected Application SHOULD gracefully handle the case when a Connection is already in use by another application.	To ensure the optimum user experience.
XAPP-0170	If the Connected Application uses data polling to retrieve data from a server then the default polling interval SHOULD be configurable by the network Operator.	To ensure optimum network utilisation and optimum battery life for the Terminal.
XAPP-0180	If the Connected Application does not require a Connection then it SHOULD release it. If the Connected Application subsequently needs a data Connection it SHOULD re-connect automatically.	Helps minimise battery drain and impact on the Operator's network.
XAPP-0190	Subject to XAPP-0360, if the user of the Connected Application User or Network Operator will benefit in terms of data speed or cost then the Connected Application SHOULD periodically check for the availability of new Connections.	Covers the scenario where a Connected Application is connected but the User walks into a WLAN hotspot.

3.3 BATTERY LIFE MANAGEMENT

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0200	The Connected Application SHOULD make the user aware of the impact that the application will have on the battery life of the Terminal (both standby time and talk time) in a user friendly manner.	Users should be made aware of how much power an application will use so that the user can make informed choices about how they use the applications installed on their Terminals. For example, an email application with a configurable polling interval could provide a user interface of the type: 'Frequent new email checking – higher power consumption' 'Once per hour new email checking – lower power consumption'
XAPP-0210	Connected Applications SHOULD make the most efficient use of the Terminal's battery as possible by implementing or using intelligent battery saving features wherever possible.	Battery life is the most precious resource on the Terminal and therefore it must be preserved as much as possible to ensure the best user experience possible. For example, a push mail application may offer the user the option to only check for email during working hours thus stopping the Terminal from polling the network during the night.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0220	Connected Applications SHOULD respect any overall system power state (e.g. “battery save mode”) but a manual override MAY be provided by a Connected Application if it is in the user’s best interest.	If a Terminal supports a global ‘battery save’ mode then the application should respect this setting. However it may be prudent for the application to provide a manual override for this setting if it is in the users best interest. For example, if the user has set their Terminal to be ‘offline’ then an email application may still provide the user with an option to manually override this setting by providing a ‘Sync Now’ button or similar.

3.4 USER EXPERIENCE

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0230	For 3G Terminals, the Connected Application SHOULD be able to remain simultaneously connected in the background whilst the User makes or receives voice calls.	3G networks allow the simultaneous transmission of voice and IP data from a Terminal. Therefore a Connected Application must still function whist the user makes or received a voice call, even if the application is put into the background.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0240	For 2G Terminals the running, Connected Application SHALL NOT terminate with an error condition due to the loss of network connectivity (e.g., because of a temporary loss of connectivity for any reason, including when a 2G Terminal is used for voice communications).	Most 2G networks cannot support the simultaneous transmission of voice and IP data from a Terminal. Voice calls take priority over data traffic and therefore an application's data Connection may be unexpectedly suspended at any time whenever the user receives and incoming call. On any network, a data Connection may be terminated unexpectedly.
XAPP-0250	For WLAN capable Terminals, the User SHOULD be able to make and receive voice calls and events like SMS whilst the Connected Application remains simultaneously connected to a WLAN network in the background	To ensure the optimum user experience
XAPP-0260	The Connected Application MUST handle out of service states (e.g. no coverage) gracefully.	The user may need to know that the Connected Application is not transacting data but beyond that any repeated Connection prompts or 'not connected' prompts add little value.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0270	Operator specific help related to the use of the Connected Application on a specific Operator's network SHOULD be provided by the application vendor.	Operators may wish to provide Operator specific help within the vendors application. A solution to this may be that the Operator gives the application developer a URL to a specific help page on their portal. Thus when the user clicks on the 'help' option within the application the application checks to see what Operator they are using (using the mobile network code) and redirects the user to the specific Operator defined URL.
XAPP-0280	A Connected Application SHOULD make the user aware of how much data the application is sending and receiving.	There are three reasons for an application to allow the user to throttle data utilisation: So that Terminal memory capacity is not impacted by application. Flat rate data charges are not universal, so network data usage must be preferably minimised. To maximise battery life.
XAPP-0290	A Connected Application MAY provide options to restrict the amount of data transmission used.	

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0300	The Connected Application SHOULD make the user aware of what services are (and are not) available when roaming.	Some services might only be available when on the users 'home' network therefore the application should make it clear to the customer that the selected roaming network has an impact on the service. For example the user may not be able to get a data Connection whilst roaming, or the user may be charged differently whilst roaming.
XAPP-0310	The Connected Application SHOULD make the user aware of what services the Connected Application can supply when connecting via different network technologies.	Some services (such as video streaming) might only be available when on a 3G network (and therefore the application should make it clear to the customer that the network technology has an impact on the service.
XAPP-0320	The Connected Application SHOULD work with the user's currently selected Theme or Profile.	The application should use the correct colour scheme, etc.
XAPP-0330	The Connected Application SHOULD define the set of services the Connected Application offers in each of the following situations: - No network connectivity available - One network technology available - Multiple network technologies available	To ensure that services offered by the Operator work in a consistent way according to network availability.

REQ. ID	REQUIREMENT	RATIONALE
XAPP-0340	Connected Applications SHOULD have the facility to send, with a user's agreement, (a provisioning or informative link to another mobile user for easy distribution of applications (e.g. via an SMS).	Social distribution of applications helps discovery. However, the user should be made aware this is happening so that the user's trust of the application is not violated.
XAPP-0350	If the Terminal offers a range of input options (e.g. T9) then the Connected Application SHOULD offer those input options.	To ensure user experience consistency
XAPP-0360	If a Connected Application is mid-session, it SHOULD NOT change Bearer (e.g. from 3G to WLAN) if there may be a negative user experience consequence in terms of billing or content availability. If it is necessary to change Bearer, the user SHALL be informed.	There could be charging or user experience consequences if the Bearer is changed without the user's knowledge, so the best approach is to avoid an ambiguous situation. For example, if a user is browsing an Operator service via a web portal that is only accessible via the Operators GPRS network it would be undesirable for the Terminal to automatically switch to using a WLAN Connection in the middle of this service.

3.5 TERMINAL REQUIREMENTS

The following requirements are intended for implementation by operating system and Terminal vendors.

Application Developers need to ensure that their applications are compatible with these requirements.

REQ ID	REQUIREMENT
XAPP-0370	A 3G Terminal MUST support simultaneous Voice and Data communications.

REQ ID	REQUIREMENT
XAPP-0380	If WLAN is supported, the Terminal MUST support WLAN whilst attached to a 2G/3G network (e.g. when browsing via WLAN incoming events like SMS and voice calls should be accepted by the Terminal and handled appropriately).
XAPP-0390	If WLAN is supported, the OS MUST be able to simultaneously support an active WLAN Connection and an active 2G/3G PDP context.

The following requirements apply to UMTS/GPRS network technologies

REQ ID	REQUIREMENT
XAPP-0400	A 3G Terminal MUST be able to support (depending on the Operator's preference) a minimum of 2 simultaneous UMTS/GPRS contexts with different APNs (i.e. primary PDP contexts).
XAPP-0410	The Terminal MUST prevent the configuration of two different APNs with the same network parameters (address and Bearer type).
XAPP-0420	The Terminal MUST allow the configuration of at least 5 different APNs
XAPP-0430	The Terminal MUST prevent the configuration of two different proxies with the same network parameters (address, ports and type)..
XAPP-0440	The Terminal MUST allow the configuration of at least 10 different proxies.
XAPP-0450	The Terminal MUST decouple APN and proxy configurations (e.g. it should be possible to use the same proxy with different APNs and vice versa).
XAPP-0460	Each packet switched Connection MUST contain an identifier, an APN and optionally one or more proxies to be used.
XAPP-0470	The Terminal MUST prevent configuration of two different Connections with the same configuration parameters (i.e. with the same combination of APN and proxies).
XAPP-0480	The Terminal MUST allow the configuration of at least 5 different Connections
XAPP-0490	The Terminal MUST support a priority mechanism for Connections.
XAPP-0500	The Terminal MUST support configuration of a default Connection

The following requirements apply to Connection Groups

REQ ID	REQUIREMENT
XAPP-0510	The Terminal SHOULD support configuration of Connection Groups
XAPP-0520	The Terminal MUST create a Connection when given a Connection Group identifier by a Connected Application
XAPP-0530	When a Connected Application tries to create a Connection using a Connections Group identifier, the Terminal SHOULD select the highest priority Connection within the group.
XAPP-0540	The Terminal MUST provide a mechanism to manage the priority of the Connections within a group (e.g. WLAN preferred over 3G).
XAPP-0550	The Terminal MUST provide an API for a Connected Application to create a Connection using a Connection identifier.
XAPP-0560	When a Connected Application requests to establish a Connection and no specific Connection is configured, the user MUST be prompted with all the Connections (and Connection Group, if supported) available so that they can decide which one to use.
XAPP-0570	When the list of Connections is prompted to the user (see XAPP-0560) the following options MUST be supported: If there is already an active Connection, the suggested (highlighted) option SHOULD be the active one with highest priority. If there is no active Connection, the suggested (highlighted) option MUST be the default Connection.
XAPP-0580	The Terminal MUST allow the network Operator to modify the Connection Policies via OTA
XAPP-0590	The Terminal MUST not allow the user to modify the Connection Policies
XAPP-0600	When a Connected Application requests a Connection that is currently in use, the existing Connection MUST be maintained and shared by the two Connected Applications
XAPP-0610	The Terminal MUST allow different Connected Applications to use the same 2G/3G PDP context (or in the case of UMTS/GPRS, the same APN) with different proxies simultaneously.
XAPP-0620	When the maximum number of simultaneous 2G/3G PDP contexts have been reached, and a Connected Application requests the use of a new Connection that requires a new PDP context (or in the case of UMTS/GPRS, a different APN) the Terminal MUST drop one of the former Connections only if its priority is lower than the new one and if the user accepts.

REQ ID	REQUIREMENT
XAPP-0630	A standardised mechanism MUST be provided to enable a Connected Application to be activated remotely by the network Operator.
XAPP-0640	A mechanism SHOULD be provided to enable Connected Applications to register their desire to make periodic use of the network.[1]

[1] For example, an API for applications to specify minimum and maximum elapsed times between periodic network usage. The mechanism should then notify registered applications when to use the network for periodic usage. For example, a mobile email client may need to periodically poll a server that does not support push email, and a mobile instant messaging client may need to periodically refresh its presence state. By co-ordinating these different applications' use of the network, the time between high-power states can be increased and hence battery life significantly lengthened. See Section 8.3 for more information.

4 DEFINITION OF TERMS

TERM	DESCRIPTION
BEARER	The physical implementation (Layer 1 in Open Systems Interconnection Model terms) of a Connection. Examples include WLAN, Bluetooth and 2.5/3G
BLUETOOTH	Bluetooth is a standard that facilitates wireless communication between devices such as mobile Terminals, PDAs (personal digital assistants) and other electronic devices.
CONNECTED APPLICATION	An application that requires direct or indirect network access
CONNECTION	Combination of one identifier, one Bearer, one APN (for UMTS/GPRS networks) and optionally one or more proxies
CONNECTION GROUP	One or more Connections collectively referred to by an identifier. For example there could be an 'Internet' Connection Group which a browser could request a Connection from.
OPERATOR	A business entity that provides communications services to customers in the form of products that include combination of Terminals and SIMs
PDP CONTEXT	A data structure holding session information about a PS Connection
PROFILE	A combination of settings selectable by the user
TERMINAL	Used as an alternative term for a cellular telephone
THEME	A set of graphics and colour definitions that customise the look and feel of the Terminal
ZIGBEE	A published specification set of high level communication protocols designed to use small, low power digital radios based on the IEEE 802.15.4 standard.

5 ABBREVIATIONS

ABBREVIATION	DESCRIPTION
3GPP	3rd Generation Partnership Project
API	Application Programming Interface
APN	Access Point Name – denotes the packet data Connection from the Terminal to the UMTS/GPRS network
CELL-FACH	Cell Forward Access Channel
CELL_PCH	Cell Paging Channel
DCH	Dedicated Channel
DL	Downlink
EDGE	Enhanced Data rates for GSM Evolution
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HSDPA	High-Speed Downlink Packet Access
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineers
IM	Instant Messaging
IMEI	International Mobile Equipment Identity is a number unique to every GSM Terminal
NFC	Near Field Communication
OTA	Over The Air
PDP	Packet Data Protocol. Any protocol which transmits data as discrete units known as packets, e.g., IP, or X.25
PS	Packet Switched. A type of network where multiple users share the same transmission channel, only transmitting when they have data to send.
QoS	Quality of Service

ABBREVIATION	DESCRIPTION
RRC	Radio Resource Control
SIM	Subscriber Identity Module
SMS	Short Message Service
SSID	Service Set Identifier
TCP	Transmission Control Protocol
UL	Uplink
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network

6 REFERENCED DOCUMENTS

NO.	DOCUMENT	AUTHOR	DATE
1	RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels	IETF	March 1997
2	“Deployment Considerations for lemonade-compliant Mobile Email" http://www.ietf.org/internet-drafts/draft-ietf-lemonade-deployments-09.txt	Randall Gellens, Qualcomm IETF	publication pending

7 APPENDIX 1 - ABOUT DATA CONNECTIONS ON MOBILE NETWORKS

When a Connected Application needs to transact data across WLAN/2.5/3G networks it requests that the Application Framework creates a Connection.

A Connection encompasses a Bearer and Bearer technology specific settings. In the case of a Connection where the Bearer is a GSM or UMTS packet switched network, the settings include an access point and optionally a proxy.

In the case of a Connection where the Bearer is a WLAN adaptor, the settings include an SSID and optionally security settings.

A Terminal may offer multiple Connections to support a range of services and billing options. The application framework selects which type of Connection to create (or re-use) based on a set of priority rules. With GSM or UMTS Packet Switched networks, a successful Connection involves attaching to the network and then activating a PDP context. The PDP context maintains session information between the Terminal and the GPRS network.

There are two kinds of PDP contexts.

- Primary PDP Context
  - Has a unique IP address associated with it
- Secondary PDP Context
  - Shares an IP address with another PDP context
  - Is created based on an existing PDP context (to share the IP address)
  - Secondary PDP contexts may have different Quality Of Service settings

Some network Operators supply two main access points – one for WAP and one for internet traffic. Furthermore, network Operators may offer dedicated access points for specific applications to facilitate value based services for their users.

Figure 1 below shows the elements involved in creating a Connection.

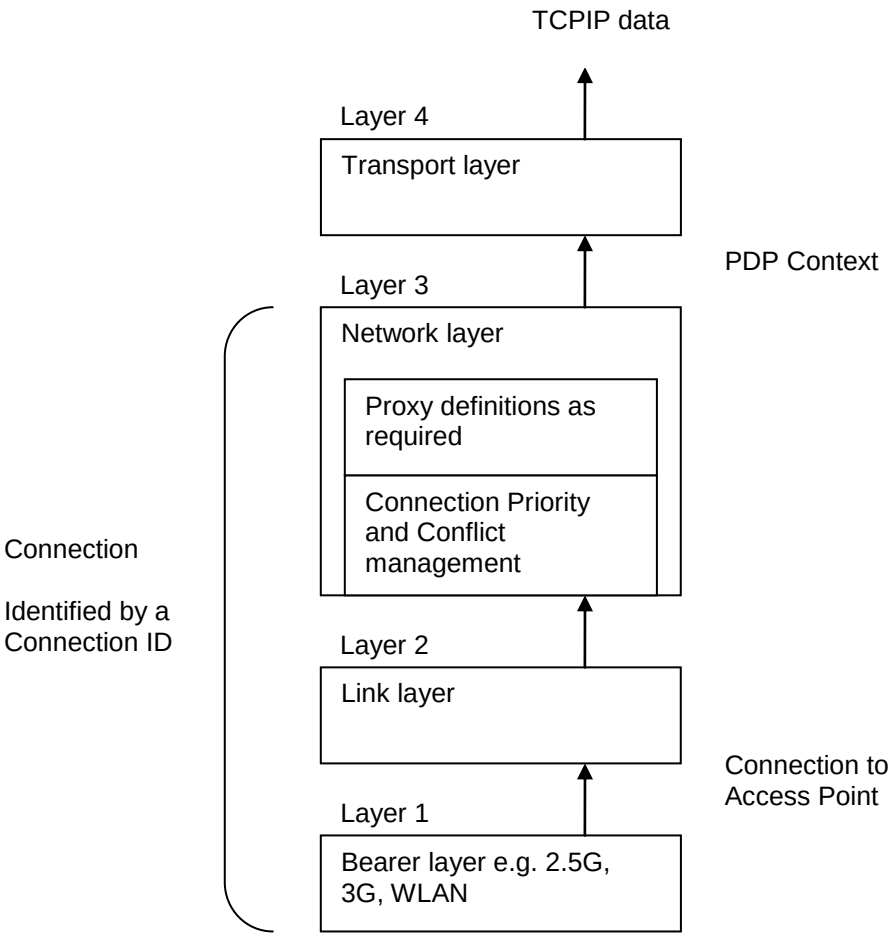


Figure 1: Elements involved in creating a Connection

8 APPENDIX 2 - ABOUT BATTERY MANAGEMENT ON TERMINALS

8.1 INTRODUCTION

The battery is the most precious resource within any mobile electronic device and the power it contains must be used as efficiently as possible by all the functions within the device. Application developers must therefore ensure that they architect their applications to be battery efficient from the outset of their development.

The aim of this appendix is to explain the underlying functions within a mobile Terminal which have most impact on battery life and, if used inefficiently by an application, can have a severe impact on battery life.

Note: This document only covers the UMTS Release 99 data Bearer as typically supported by all UMTS Terminals today. This document therefore doesn't describe the behaviour of UMTS Release 6 HSDPA data Bearers.

8.2 KEY DIFFERENCES BETWEEN UMTS AND GSM RADIO BEHAVIOUR

- A UMTS radio take 5 seconds on average to transition from standby to active mode and could take anything from 15 to 100 seconds to transition from active to standby mode depending on both the capability of the radio network equipment and the values defined by the Operator for their transition timers.
- When in active mode a UMTS radio typically receives data at a rate between 100 and 340 kbps and transmits data at a rate between 45 to 100 kbps. Typical average data rates are 128kbps in the downlink (network to mobile) and 64kbps in the uplink (mobile to network). It is worth noting that these are the bit-rates at the radio layer, the net bit-rate at the application layer will be less once the effect of adding IP headers and other lower layer data wrappers are taken into account.
- A GSM radio takes 1 to 2 seconds to transition from standby to active mode and 1 to 2 seconds to transition from active to standby mode.
- When in active mode a GSM radio typically receives at between 15 and 35 kbps and transmits at between 5 to 20 kbps (depending on the number of GSM timeslots the network Operator allows the Terminal to use). If the GSM network supports EDGE the bit rate can be even higher - around 80kbps for the downlink and 40kbps for the uplink.

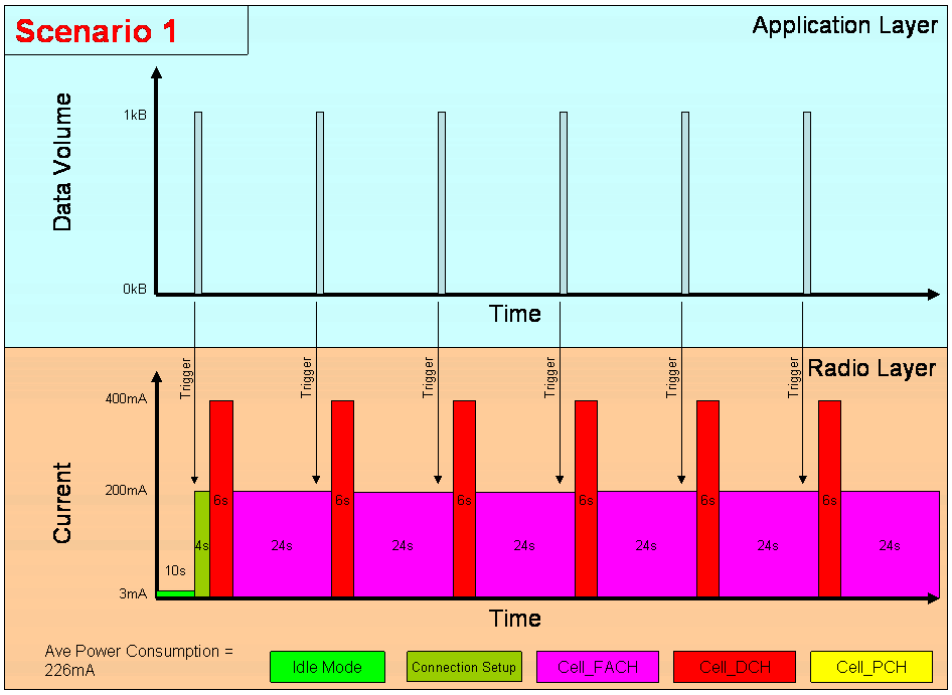
8.3 UMTS EXAMPLES

The following four scenarios show some examples of the power consumed for three different application behaviours. The top part of each graph shows the data being sent at the application layer in the Terminal. The bottom part of the graph shows the behaviour of the radio within the Terminal. As can be seen,

the events at the application layer are triggering the radio to transition between its various states. What should be noted is that the different states consume significantly different amounts of power. For example when the radio is in DCH (Dedicated Channel) mode it consumes about 400 milli-amperes compared to idle mode which only consumes about 3 milli-amperes. So the radio, when in DCH mode, consumes 133 times the power than it does in idle mode! Keeping the Terminals radio out of DCH mode and in idle mode must therefore one of the key design criteria for any application developer.

In the first scenario we have an application sending data to a server every 30 seconds. This behaviour keeps the UMTS radio in DCH and FACH modes, resulting in a high average power consumption of 226mA. If the application keeps sending data like this ad-infinitum, the typical battery life of the Terminal would be reduced to 1200 mAH ÷ 226mA which equals 4.8 hours. As can be seen, this type of behaviour by an application should be avoided at all costs when using a UMTS radio.

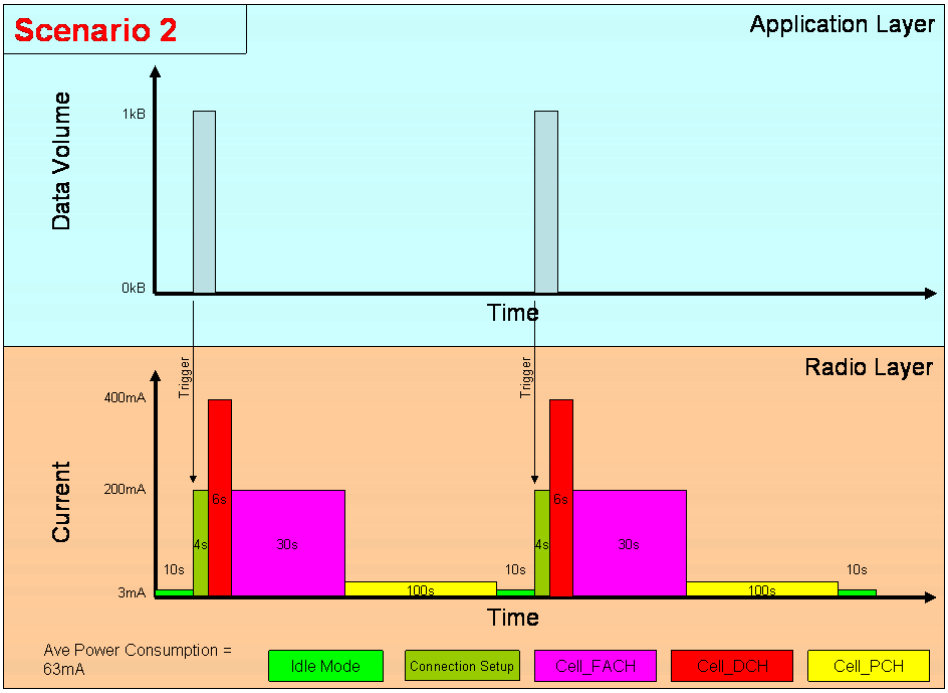
It should also be noted that these examples are equally applicable to the use of downlink data on a UMTS network - i.e. if the top graph represented the volume of downlink data being pushed to a Terminal then there would be exactly the same response at the radio layer and, more importantly, the Terminal would consume the same amount of power.



In the second scenario, we have an application that sends data to the server every 160 seconds. Each time the application sends data it causes the UMTS radio to transition through its various states, but in this case the gap between the data transmissions is sufficiently long enough for the UMTS radio to

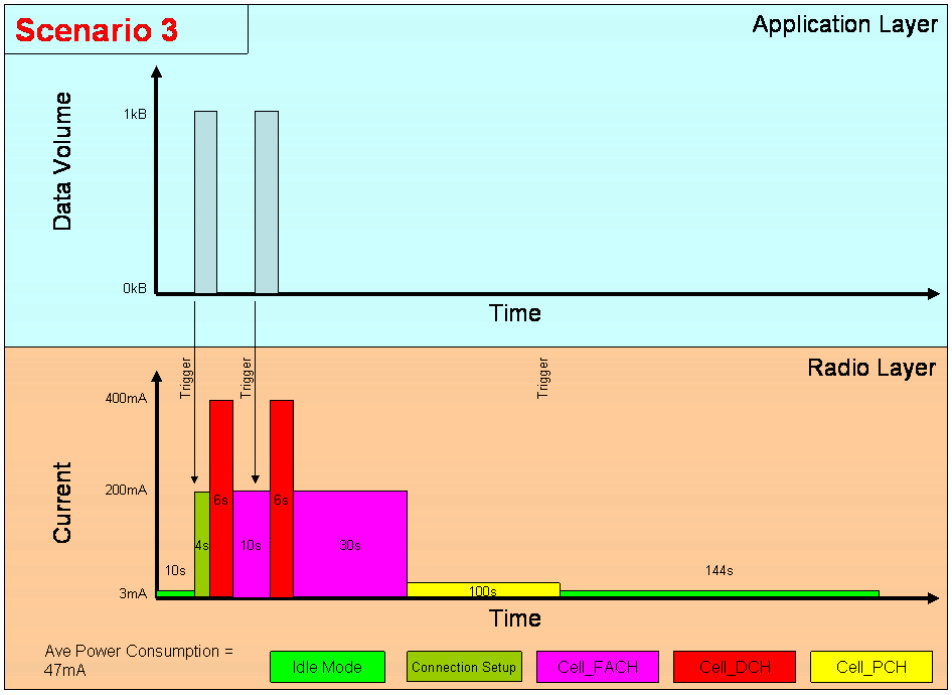
transition to its more power efficient states (the CELL_PCH and Idle modes). This reduces the average power consumption to 63mA.

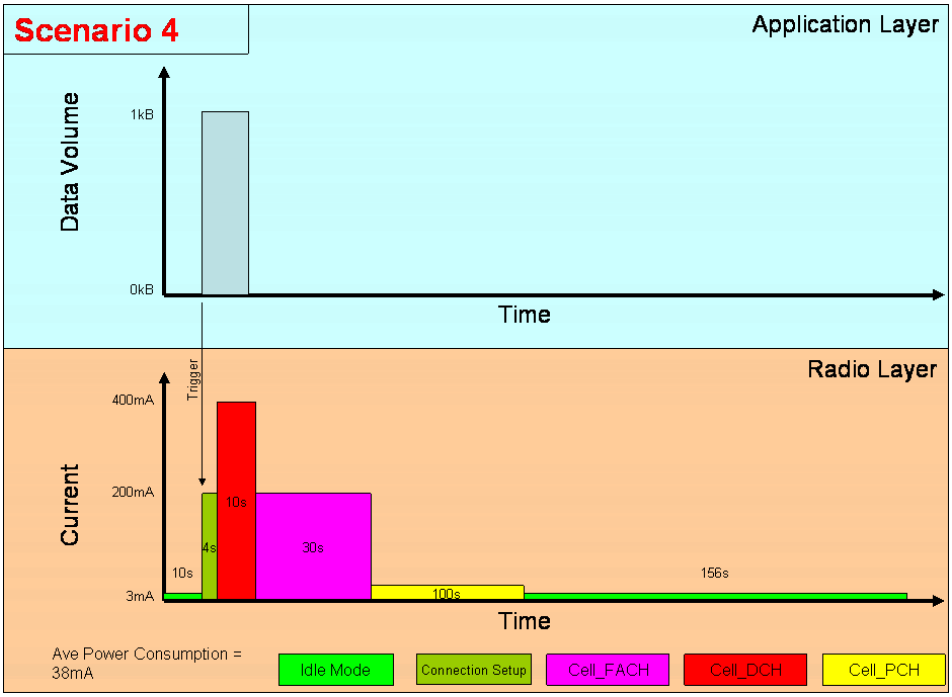
If the application keeps sending data like this every 160 seconds, the battery life of the Terminal would be 1200 mAH ÷ 63mA, which equals 17 hours. Note that in this scenario, the volume of data sent by the application is the same as in the first scenario, but in this case the effect on battery life is increased by a factor of approximately four.



Scenarios 3 and 4 improve the battery life performance yet again. In these two scenarios the data in sent in two short bursts (in scenario 3) and one longer burst (in scenario 4). If these data transmission Profiles were to be repeated by the application ad-infinitum, then the average power consumption would be 47 mA for scenario 3 and 38mA for scenario 4. This would yield a batter life of 23 hours and 28 hours respectively.

Note, however, that even these scenarios still drain the battery too fast and have too much of an impact of the standby time of the Terminal. To reduce the current drain caused by an application which periodically sends or receives data from the network to an acceptable level we have to increase the time between data sends/receives into the order of hours rather than minutes!





The conclusions we can draw from these examples are:

- There is a massive difference in the power consumed whilst the Terminal is in idle and Cell_PCH modes compared to when the Terminal is in DCH, Cell_FACH and Connection set-up modes.
- Applications should always aim to keep the Terminal is its low power mode for as long as possible.

- Applications should do this by sending larger amounts of data infrequently to the network, as opposed to sending small amounts of data to the network very frequently.
- The time period between data sends needs to be larger than might be expected to preserve an adequate standby time on the Terminal. Even for scenario 4, which only sends a burst of data to the network every 310 seconds the impact on standby time is too great for most users. The time period between such transmissions therefore needs to be in the region of 30 minutes to an hour to adequate preserve battery standby time. (If the duty cycle for scenario 4 were increased from 310 seconds to 30 minutes, the average current would be 9mA which would yield a standby time of 120 hours).
- Developers of Connected Applications which have to periodically send or receive data from a network should also bear in mind that a user may want to run multiple Connected Applications on their Terminal at once. In this case it is important that application vendors take genuine measures to minimise the power consumed by their application.
- When there are multiple Connected Applications that periodically send or receive data using the network, battery life can be significantly extended if the applications are able to coordinate their network use and make opportunistic use of an existing connected state. Multiple applications cooperate by doing their periodic data bursts at the same time or immediately after each other, in order to avoid cycling through radio states at shorter intervals. To accomplish this, it is highly desired that the Terminal operating system or application framework provide a mechanism for Connected Applications to register their interest in using the network. For example, an API that allows applications to indicate a range of time for periodic network use, such as between 20 and 40 minutes, or between 30 and 90 minutes, etc. The Terminal operating system or application execution environment can then co-ordinate network use by notifying the registered applications when it is a good time to use the network. This could be when the radio is in a high-power state but the network is idle, for example (such as between page loads by a web browser, or after termination of a voice call). Even simple co-ordination, such as notifying registered applications in a round-robin fashion during their time period, can have significant impact on battery usage.

8.4 Typical Rules and Timings for Radio State Transitions:

The table below gives some typical criteria and timings for the radio state transitions. It should be stressed that these are only indicative values – in reality each Operator's network will be configured differently depending upon their radio network optimisation strategy and their choice of radio network vendor.

TRANSITION	CRITERIA	TIME
IDLE TO DCH	There is more than 256bytes in the buffer for >20mS	4 seconds
DCH TO CELL_FACH	There is no data in the buffer for 5 seconds.	5 seconds
CELL_FACH TO DCH	There is more than 256bytes in the buffer for >20mS	1 second
CELL_FACH TO CELL_PCH	No data in the buffer for 30 seconds	30 seconds
CELL_PCH TO IDLE	No data in the buffer for 100 seconds	100 seconds

8.5 *TYPICAL POWER CONSUMPTION FOR THE DIFFERENT RADIO STATES:*

The table below gives some typical figures for Terminal power consumption in the different radio states. The actual values for these figures varies depending upon the make of Terminal, network Operator settings and radio conditions. As of January 2007 the power consumption for most Terminals will be within the range stated below. These figures will undoubtedly reduce over time as Terminal technology develops.

STATE	TYPICAL POWER CONSUMPTION
IDLE	4mA to 6mA
DCH	250mA to 400mA
CELL_FACH	200mA to 250mA
CONNECTION SET UP	200mA to 250mA
CELL_PCH	6mA to 12mA

8.6 *BATTERY FAQ'S*

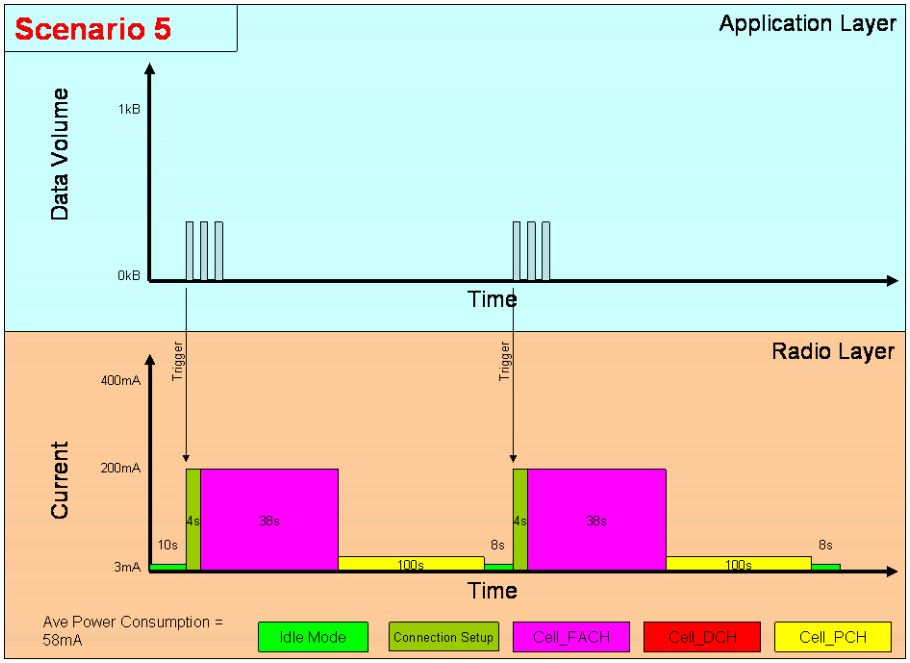
Q. Some radio network equipment vendors tell me that if my application only sends a small amount of data (<256bytes), or sends data very slowly (~3kbps) then battery life will be improved. Is this true?

A. There is some truth in this, however the gains are, in practice, likely to be small. The reason some network vendors say this is that sending small amounts of data or sending data very slowly will keep the Terminals radio in Cell_FACH state rather than in DCH state. When in Cell_FACH state, the Terminal can send data to the network but only at a very slow rate. The battery saving occurs because the Terminal consumes less power in Cell_FACH sate than in DCH state. An example of this scenario is shown below.

In scenario 5 the application is periodically sending the same volume of data as in scenario 2, but in this case it is sending the data to the network in 3 very small packets as opposed to 1 larger packet. This has the effect of ensuring the Terminal only ever transitions to the Cell_FACH state. The net result is that the average power consumed in this scenario is 58mA, compared to scenario 2 where the average was 63mA, yielding a saving of 5mA on average. This saving will probably not in practice make that much difference to the overall battery life of the Terminal.

Another issue with this approach is that each radio network vendor) and Operator define different criteria for the transition to DCH mode. This means that the application developer might successfully be able to implement this power saving technique on one type of network, but on another network it will not work because the state transition criteria are set differently!

Another problem for the application vendor is that the transition to DCH is also based upon the data throughput at the radio layer and not the application layer. This means that an application, that sends 100 bytes of data at the HTTP layer will be wrapped in IP headers and various other lower layer data headers as it passes down though the protocol stack in the Terminal. So what started out as a 100 byte data message may end up as a 300byte data message being sent over the radio! Therefore much lower data rates than might first be expected must be applied by the application developer to ensure the Terminal does not switch into DCH mode.



Q. Why are the state transition timers set so long on a UMTS network? Why for example don't the network Operators reduce the timer for Cell_FACH to Cell_PCH to, say, 2 seconds?

A. Many Operator's have optimised their data networks for web browsing and this has dictated the timer values. This is because the data profile whilst web browsing typically causes data peaks every 10 to 20 seconds as the user browses from page to page. This profile means that many Operators choose to keep the Terminal in Cell_FACH mode for typically 30 seconds in preparation for the next page download request to arrive. They keep it in Cell_FACH because it is quicker to transition from Cell_FACH to DCH (approx 1 second) than it is from Idle to DCH (approx 5 seconds),giving a better user experience whilst browsing.

Unfortunately, at the present time, these timer vales are not application-aware, so the same timer is applied to all applications.

As other types of Connected Applications become more common-place, Operators and network vendors will start to optimise their networks and Terminals to perform more efficiently with these different data services. This, however, remains a difficult task whist data Bearers remain non-application aware.

Q. Why does is take so long to transition from Idle to DCH?

A. This delay is inherent to UMTS networks and is unfortunately something that cannot be changed by the Operator. There is an activity within 3GPP (the

standards body that defines the UMTS standard) to shorten the RRC Connection set-up and the radio Bearer Set up procedures. However, it may be several years before these techniques are commonly deployed by the network Operators.

Q. Why do UMTS radios consume so much power?

A. Most of the power is consumed within the UMTS power amplifier, which even to day, are only 30 % efficient at best (i.e. the power transmitted out of the antenna is only 30% of the power that goes into the power amplifier chip!). The reason for this is that UMTS radios require very linear, wide-band, power amplifiers and the only ways to achieve this today all incur such high power losses. The efficiencies of these power amplifiers are getting better as time goes on, however the improvements are progressive and no 'step change' in efficiency is expected.

Typical Power Consumption Figures for a set of 'class leading' 3G Handsets:

Tx Power		Vendor A	Vendor B	Vendor C	Vendor D	Vendor E
Live Cell		287 mA	351 mA	301 mA	281 mA	240 mA
Transmit Power	0 dBm	290 mA	354 mA	300 mA	302 mA	244 mA
	-5 dBm	288 mA	350 mA	310 mA	290 mA	236 mA
	-10 dBm	284 mA	340 mA	306 mA	285 mA	230 mA
	-15 dBm			304 mA	284 mA	228 mA
	-20 dBm	280 mA		303 mA	284 mA	227 mA
	-40 dBm	278 mA			280 mA	223 mA

The above measurements were performed using Terminals attached to a Rohde & Schwarz CMU200 UMTS test-set.

Q. What is the relationship between uplink and downlink data-rates and does this have any impact on battery life?

A. Uplink and Downlink data-Bearers are tied together in UMTS systems. The typical combinations are 64kbps UL / 64kbps DL, 64kbps UL / 128kbps DL,

64kbps UL / 384kbps DL, 128kbps UL / 128kbps DL, and 128kbps UL / 384kbps DL. The network decides which data Bearer to give a Terminal based upon the available network capacity, radio signal quality, Terminal capability and application data demand. Typically the network will adjust the data Bearer 'real time' as and when any of these parameters changes. It is therefore quite common for the data Bearer to change from 384kbps in the downlink to 128kbps as the radio signal quality deteriorates, for example, when the user travels into an area of weak signal strength or the during a time when the network is heavily loaded with data traffic.

Generally speaking, the data Bearer being used makes very little difference in the amount of power consumed by the Terminal. This is because the transmit power from the Terminal will remain about the same regardless of the uplink data being sent and the receiver works just as hard to receive a 64kbps Bearer as it does for a 384kbps Bearer.

Q. Do networks support QoS and does using QoS have any effect on battery life?

Very few (if any) UMTS network support QoS parameters for data Bearers. Therefore, as of today, if an application requests a Bearer with particular QoS it will usually be ignored and the network will issue the Terminal with a 'background' class data Bearer.

Many networks will support QoS in the future and so it is worth it for application vendors to make use of QoS parameters today if they can, so that they can potentially make use of QoS Bearers in the future. It's worth noting, however, that it is the network Operator that controls which applications can have QoS Bearers so application vendors may have to work closely with Operators to guarantee that their applications will get QoS Bearers. Non Operator-approved applications that request Q0S Bearers are likely only to be issued background class data Bearers regardless of what QoS they request.

-------- END OF DOCUMENT --------