Telstra Wireless Application Development Guidelines

Version 10 Issue 1

Date: September 2019

TABLE OF CONTENTS

1. AIM

The goal of this document is to:

1. educate developers about Telstra's mobile network;
2. encourage developers to produce network "impact friendly" applications i.e. applications that don't impose an unnecessary strain on the mobile network; and
3. encourage more efficient use of the Telstra mobile network by developers, in order to enhance customers' experience of the Telstra mobile network.

It is important to understand how to best develop for our mobile network. Your design choices can have significant effects on how well your application works, as well as potentially give your product or solution a competitive edge and improve its performance, by improving things like its feature set, longevity and compliance to industry standards.

2. SCOPE

These guidelines cover application development for smartphones, tablets and laptops using Telstra's mobile network. They also cover Machine to Machine (M2M) and Internet of Things (IoT) communication applications including embedded modules, devices integrating embedded modules and the related controlling software behaviour.

For those requiring additional detail on each topic, links are provided to explore further.

The scope of the document excludes the details of how to code applications and associated backend servers / cloud. It does not cover details of user interface visual design or programming languages.

This document draws from a wealth of existing industry associations, OS platform, wireless operator and other developer guidelines.

3. DISCLAIMER

These guidelines are general in nature and apply to the most common use-case scenarios. You should consider your own specific requirements where necessary. Reliance upon any representations made or information contained in this document is at your own risk.

Telstra will seek to update this document periodically. Please check the Telstra.com website for an updated version. If you are building solutions specifically for Telstra or other customers seek guidance to ensure your solution delivers the product's desired performance.

4. INTRODUCTION

4.1. Why are the guidelines required?

Wireless Network Constraints
Wireless networks have some constraints that developers need to consider when developing their applications such as:

Power Source – Wireless devices typically rely on batteries whereas home/fixed network routers are AC powered.

Reliability/Robustness of network connection – Wireless networks deliver varying throughputs based on variables such as radio conditions.

Network Technology Speed and Latency Variability – Cellular throughputs vary considerably based on location, terrain, coverage, radio interference, geography, and technology. The latency of 4G is several times faster than that of 3G for example.

Capacity constraints – Apps must be designed to work well, and be responsive even when networks are heavily loaded.

4.2. Benefits of the guidelines?

Following the principles in these guidelines will provide benefits for the developer, app user and the network operator. Such benefits include:

- improved battery life for devices;
- lower data costs – if an application uses the network efficiently, it will use less data; resulting in lower data costs for the app user in connection with that application;
- more responsive apps;
- increased application / device longevity;
- more robust / resilient applications;
- reduced network signalling; and
- improved user security and privacy.

Section 7 below (Application Development Techniques) sets out the techniques which developers can use to deliver the above benefits.

5. AN OVERVIEW OF THE TELSTRA NETWORK

5.1. Technologies and Features

Mobile networks evolve rapidly. Developers need to ensure they future proof their design and plan to take advantage of new features as they become available.

The following table presents the technologies and features available on the Telstra Network (as of September 2019) along with future technology trends. Included alongside are relevant considerations for designing your wireless application.

Technology/Feature	Present	Future	Considerations
3G Frequency Bands Uses HSPA and HSPA+ technologies	• 850 MHz (B5)	• 3G currently has a large geographical footprint, no network expansion is planned.	• No significant 3G design feature development is planned.
4G Frequency Bands Uses LTE technology	• 1800 MHz (B3) for coverage • 700 MHz (B28) for coverage • 2600 MHz (B7) for capacity and in-building coverage • 2100 MHz (B1) for capacity	• 900 MHz (B8) and 850 MHz (B5) may be supported in selected areas	• Note that the 700 MHz bands (B12, 13, 14, 17) used in the US are not compatible with the Australian 700 MHz band (B28) • Lower frequency bands work better in rural areas. • Refer below for coverage map *
LTE Carrier Aggregation (CA) Combines 2 or more carriers together to allow greater throughput	Support 2, 3, 4 and 5 CA combinations of bands 1, 3, 3, 7, 7 and 28	• 6 Band CA: 1 + 3 + 3 + 7 + 7 + 28	• CA offers higher data rates that are suitable for large file downloads and video streaming applications • CA is limited to certain areas of the network
MIMO Multiple In Multiple Out. Using multiple antennas for the up and downlink of radio transmission can increase throughput or received signal quality	In downlink network supports: • 2x2 on all bands • 4x2 on B1, B3 and B7 • 4x4 on B1, B3 and B7	MIMO support is in plan to be introduced on the following bands: • 4x2 on B28	
LTE-M/NB-IoT Low throughput and low power cellular technology for IoT solutions	• 700 MHz (B28) for Cat M1 and NB1	• 1800 MHz (B3) for Cat M1	
IPv6 The most recent version of the internet addressing protocol. It has a far larger number of unique addresses than IPv4	• IPv4 • IPv4v6 DS deployed on telstra.internet APN for mobile broadband • IPv4v6 DS deployed as default configuration but IPv6 SS is preferred for IoT devices	• In future it is expected that all devices shall support 464 XLAT and IPv6 SS • Devices shall be configured as IPv6 SS on telstra.wap	• APNIC no longer has any IPv4 addresses and the industry is moving to IPv6

*Telstra LTE coverage footprint: https://www.telstra.com.au/coverage-networks/our-coverage

5.2. LTE Device Category Network Support

"Device categories" refer to different levels of maximum theoretical data rates supported by devices. Devices will have a maximum category that they can support. A throughput speed within that range can only be achieved if the network supports the same maximum category and is also dependant on network radio conditions.

The following table lists the most common LTE categories currently supported by the Telstra Network. This is not an exhaustive list. If your device supports a category not specifically mentioned in this table please contact Telstra so we can help you determine if that device category is supported on our network.

The new categories (Cat 1, M1, NB1) have been designed specifically for IoT (refer to section 5.3. A key difference between these and traditional LTE categories is that capability set has been modified to enable lower power consumption, reduced device complexity and lower cost.

A summary of the key LTE categories supported on the Telstra network:

Category	Downlink (max)	Uplink (max)	3GPP Release*
NB1	100 kbps	100 kbps	Rel. 13
M1	1 Mbps	1 Mbps	Rel. 13
1	10 Mbps	5 Mbps	Rel. 8
3	100 Mbps	50 Mbps	Rel. 8
4	150 Mbps	50 Mbps	Rel. 8
6	300 Mbps	50 Mbps	Rel. 10
9	450 Mbps	50 Mbps	Rel. 11
11	600 Mbps	50 Mbps	Rel. 11
13	390 Mbps	N/A**	Rel. 12
15	800 Mbps		Rel. 12
16	1050 Mbps	N/A**	Rel. 12
13	N/A**	150Mbps	Rel. 12
18	1200 Mbps	N/A**	Rel. 13
19	1650 Mbps	N/A**	Rel. 13
20	2 Gbps	N/A**	Rel. 14
21	1400 Mbps	N/A**	Rel. 14

*3GPP release refers to the release version of industry standard covering the device category.

** Note that as of 3GPP Release 12 the uplink and downlink category speeds have been split so that they can be paired in different combinations. This means that an area which supports a particular category's downlink speed doesn't necessarily support the same category uplink speed. Therefore UL speed is dependent on the UL category and is independent of the DL category.

5.3. Internet of Things (IoT)

5.3.1. Introduction

The Internet of Things is an emerging technology trend based heavily on the M2M (Machine to Machine communication) market. IoT systems will typically comprise of many (typically hundreds and even thousands) low cost and low power devices which communicate with other devices, microservices and user applications across networks via cloud servers. These devices will tend to serve a singular function that requires very little data transfer rates and data usage.

For this reason there have been new LTE device categories created to address the Internet of Things. These new categories focus on addressing the previously unmet needs of these IoT solutions by providing low throughputs that require less energy consumption and extend the effective coverage area.

Telstra recommends that developers use LTE technology for their IoT solutions. LTE supports the newly developed IoT specific device categories (Cat 1, Cat M1, and Cat NB1).

5.3.2. Power Saving Mode (PSM)

Power Saving Mode is a feature designed for IoT devices to assist them to conserve battery power and potentially achieve a 10 year battery life.

While it has always been possible for a device to turn its radio module off to conserve battery power, the device would subsequently have to reattach to the network when the radio module was turned back on, the reattach procedure consuming a small but finite amount of energy. The cumulative energy consumption of reattaches can become significant over the life of a device.

When a device initiates PSM with the network, the network retains state information and a reattach procedure is not required, even if the device awakes and sends data before the expiration of the time interval it agreed with the network.

As an example for a monitoring application, the device might be configured by an application to enable PSM, negotiate a 24 hour time interval with the network and provide a daily status update to a centralised monitoring point. If the device's monitoring application were to detect an alarm condition, irrespective of any agreed sleep interval, the application could wake the radio module instantly and send vital information without the need to for a reattach procedure.

In a similar manner to a radio module that has been powered off, a radio module with PSM enabled cannot be contacted by the network whilst it is asleep. The inability to be contacted whilst asleep may preclude the use of PSM for some applications.

5.3.3. Extended Discontinuous Reception (eDRX)

Extended Discontinuous Reception is an extension of an existing LTE feature which can be used by IoT devices to reduce power consumption.

Today, many smartphones use discontinuous reception (DRX) to extend battery life between recharges. By switching off the receive section of the radio module for a fraction of a second, the smartphone is able to save power. The phone cannot be contacted by the network while it is not listening but if the period of time is kept to brief, the phone user will not experience degradation of service. E.g. If called, the phone might ring a fraction of a second later than if DRX was not enabled.

eDRX allows the time interval during which a device is not listening to the network to be greatly extended. For an IoT application it might be acceptable for the device to not be reachable for a few seconds or longer. Whilst not providing the same levels of power reduction as PSM for some applications eDRX may provide good comprise between reachability and power consumption.

5.3.4. Enhanced Coverage

Some IoT applications require devices to be positioned in in very poor radio conditions where the signal is extremely weak. For example, underground parking garages and in ground pits. The 3GPP Enhanced Coverage feature has the potential to increase the depth of radio coverage to enable IOT devices to be placed and operate in locations that would otherwise not be possible.

The 3GPP Enhanced Coverage feature, also called Coverage Extension, increases the power levels of signalling channels together with the ability to repeat transmissions. Through repeated transmission the ability of receivers to correctly resolve the message sent is improved. The trade-off is that repeating signal transmissions consumes additional power and the time between battery recharge or replacement may be reduced.

5.3.5. IoT Typical Usage by LTE Category

The following table is intended to help developers determine the most suitable device category for their IoT solution. You should choose a suitable device from the most appropriate device category to support the characteristics of your specific application.

LTE Category	> = 13	> = 1	1, M1	1, M1, NB1	M1, NB1
Device Data Usage **(UL+DL bytes per day)**	> 10 MB	> 1 MB	0.1 – 1 MB	< 0.1 MB (100 kB)	< 0.01 MB (10 kB)
LTE Bands Required	Triband (B28, B3, B7)	Dual Band (B28, B3)	Dual Band (B28, B3)	Dual Band (B28, B3)	Single Band (B28)
Typical Use Cases	• Video streaming • Connected Car	• Connected Home • Wearables	• Logistics • Remote Healthcare	• Smart City • Energy Metering	• Environmental Monitoring • Industrial Sensors

5.3.6. Telstra Connection Management

Telstra offers connection management platforms which allow IoT/M2M business customers to easily manage their IoT/M2M deployments. These are cloud services with a web based dashboard interface, such as the Cisco Control Centre, which provide users with a single access point from where they can:

- Activate and deactivate SIMs/services
- Run near real-time diagnostics and get diagnostic alerts
- Review connection session history and near real-time connectivity status
- Control data usage costs
- Receive SMS or voice messages from your devices to the Control Centre. Send SMS or voice messages from the Control Centre to your devices to trigger or respond to events
- Set business rules and customised alerts to identify abnormal activity or device failures
- Run customised reports to get service and usage statistics
- Use the Control Centre API to interface with external server applications to automate tasks

The Control Centre has an optional Location Service, which provides a capability to monitor location of devices. This feature does not require the device to support GPS and instead makes use of the Telstra mobile 3G/4G mobile network. It can act as a backup service in case of GPS failure to locate and detect movement of devices across cell towers.

More information on Telstra's Connection Management Platforms can be found at:

https://www.telstra.com.au/business-enterprise/products/internet-of-things/capabilities/connection-management-platform

5.3.7. Telstra IoT Platform

The Telstra IoT Platform is a cloud based service with device management, data collection, visualisation and analytics capabilities. The IoT Platform paired with Telstra certified devices and Telstra mobile network connectivity services can be used to build an end to end IoT solution. The IoT Platform provides users with the ability to:

- Manage device configurations, settings and software/firmware updates Over the Air
- Manage and store incoming data streams from client applications on deployed devices/sensors
- Analyse and visualise data to generate actionable insights and business intelligence
- Integrate Platform services with external applications/microservices using RESTful APIs

More information on the Telstra IoT Platform can be found at:

https://www.telstra.com.au/business-enterprise/products/internet-of-things/capabilities/cumulocity

The Telstra IoT Platform is built on top of Cumulocity IoT. The below listed guides walk developers through steps to integrate their IoT systems with the Cumulocity platform. Developers should consider using modules with native support for LWM2M or MQTT protocol stacks to simplify device integration with the IoT Platform.

- General guide on Cumulocity's RESTful API: https://cumulocity.com/guides/reference/rest-implementation/
- Guide on developing device client applications for interfacing with Cumulocity (using MQTT): https://cumulocity.com/guides/device-sdk/introduction/
- Guide on developing external microservices/server applications for interfacing with Cumulocity: https://cumulocity.com/guides/microservice-sdk/introduction/

5.4. Telstra 2G and 3G Technology Closure

Telstra 2G network was shut down on the 1st of December 2016. Existing 2G-only products (handsets, IoT/M2M applications, etc.) ceased to have mobile connectivity from this date. If they have not already, these applications will need to migrate across to devices that can support 4G technologies.

No formal date has been set for the closure of Telstra's 3G services. We expect it to remain in service for quite some years yet. When a decision is made, Telstra customers with 3G services will be given significant notice to ensure the transition is as seamless as possible.

To provide IoT/M2M devices with the longest possible support, Telstra recommends using the most recent technology available. Telstra considers LTE (4G) the technology of choice for IoT/M2M applications because it:

- Has a longer life expectancy
- Can support more devices per unit area
- Supports new IoT/M2M specific device categories (see section 5.3) that have better power consumption, lower cost and better coverage characteristics

5.5. Telstra Certified Devices

Telstra believes in customer first and puts the customer at the centre of everything we do. The Telstra certification and testing program is designed to make sure that your device is compatible with the network.

Telstra certification will help ensure that the device meets requirements on:

- Frequency band support
- Data throughput performance across all networks including 3G, 4G and Wi-Fi if applicable
- Network interoperability under stationary and mobile conditions
- Device performance under congested network environment
- Network reacquisition and retry algorithms
- Data and device stability
- Radio compliance
- Antenna sensitivity
- Over the air firmware and application upgrades
- IPv6 functionality
- Battery life for low power devices

The following guidelines outline the process to have your device tested and approved by the Telstra M2M Device Certification Program: https://www.telstra.com.au/content/dam/tcom/business-enterprise/machine-to-machine/pdf/business-enterprise-m2m-device-certification.pdf

Telstra recommends integrating a certified module in your end use/finished device, which will significantly expedite the testing process. A list of Telstra certified IoT/M2M modules and devices can be found at: https://www.telstra.com.au/content/dam/tcom/business-enterprise/machine-to-machine/pdf/telstra-m2m-certified-devices-modules.pdf

6. CONSIDERATIONS FOR DEVICES INTEGRATING MODULES

6.1. Appropriate Technology Choices

6.1.1. Description

When selecting an embedded module for an integrated IoT/M2M device, a developer should take into consideration the module's supported cellular features and capabilities.

6.1.2. Methods

Appropriate Cellular Technology Choice

Telstra recommends LTE as the preferred technology. LTE only modules are available as well as multimode LTE + 3G modules. Telstra will not certify 3G only modules.

Coverage / Radio Network Technology Support / Frequency Band Support

It is important to ensure that LTE coverage is available over the entirety of the expected usage areas. Telstra's LTE coverage is constantly expanding so developers should refer to our coverage maps for the latest information at: https://www.telstra.com.au/coverage-networks/our-coverage

Refer to the table in section 5.1 of this document for information regarding Telstra's current and future frequency band support for both 3G and 4G technologies.

Throughput performance

Choose a module based on your IoT/M2M solution's throughput and data usage requirements. See table in section 5.3.5 for guidance on choosing the appropriate LTE category based on typical IoT use cases. As a rule the higher the data throughput usage the faster category should be selected to reduce time connected to the network which is the highest device power consuming state, and improves end user experience.

Choose devices /modules certified for use on Telstra's network

Certified modules have been tested by Telstra for compatibility with Telstra's network. Approved modules and devices will typically have better longevity due to greater compatibility with our networks - features, technology and frequency bands. Non-approved modules and any devices integrating them, have no guarantee that they will work with our network currently or into the future. For IoT/M2M Integrated Device approval, there is a streamlined process for devices integrating already approved Modules

Data only or Voice and Data

Note when selecting an embedded module - some modules only support data and others support both voice and data. If the application doesn't require voice, then a data only module is recommended as it will be cheaper and less complex.

FOTA (Firmware over the air)

FOTA is a requirement for new modules to be certified for use on our network. Having the ability to fix bugs and update devices remotely saves customers a lot of time and energy down the track. Particularly for an IoT solution that may feature hundreds or even thousands of devices spread out over a large area it would not be practical to physically attend to each device individually to install a software patch or update.

Module Radio Network Features

Choose modules that support important radio network features such as:

- **CDRX** (Connected Discontinuous Reception) – this is a device feature that allows the device to have micro sleeps. This feature allows the device to reduce battery consumption while minimizing impact to latency. CDRX is only invoked after 100ms of inactivity.
- **eDRX** (extended Discontinuous Reception) – this is an IoT device feature that requires network support. It allows the time interval during which a device is not listening to the network to be greatly extended, reducing battery consumption.
- **PSM** (Power Save Mode) – this is an IoT device feature that requires network support. The device will inform the network that it will be going into power save mode (using close to no power in this state) along with information about when it will 'wake up' for a short period to receive any messages that may be waiting.
- **RAI** (Release Assistance Indicator) – this is an IoT device feature that requires network support. It allows the device to indicate to the network that it is not expecting to receive/transmit any more uplink/downlink data. Upon receiving this signal the network moves the device into an idle state immediately rather than relying on a (10 seconds long) inactivity timer, reducing battery consumption.

6.2. Regulatory Issues

When integrating a module into a device, regulatory requirements need to be met. These are captured by the ACMA RCM - Regulatory Compliance Mark for the product (and embedded module). The RCM indicates a device's compliance with applicable ACMA technical standards — that is, for telecommunications, radio communications, EMC and EME.

Refer: https://www.acma.gov.au/Industry/Suppliers/Product-supply-and-compliance/Steps-to-compliance/product-labelling

6.3. Antennas

- 3GPP defines how many antennas each LTE category shall support, and devices shall comply to these requirements. Reducing the number of antennas has an negative impact on the received signal which impacts performance and customer experience
- Antennas should support all frequencies bands supported by both the module and network
- Antenna should be optimized to suit the frequency bands to be used by the device.
    - For 3G devices they should be optimized for band 5 (850 MHz)
    - For 4G devices they should be optimized for all the bands they support and particularly bands 3 (1800 MHz) & 28 (700MHz)
- When installing remote equipment, directional antennas should be oriented toward the strongest received signal (in most cases)
- Antennas should be configured and placed to optimise radio performance within the physical constraints of the end product. Mounting location and space allocation should be considered in the early phases of the product development process in order to maximise performance.
- Pattern shape is another antenna performance parameter that should not be overlooked. Omni directional type pattern is much more desirable than the directional one. Parasitic coupling to metal structures near the antenna can alter its pattern shape and operational bandwidth.
- To minimise interference, it is recommended that the positions of the antennas are as far as possible from any digital circuitry that generates high frequency noise (that is, high speed clocks).

6.4. RF Shielding / Interference Mitigation

Integration of wireless modules into end products shall minimise any possible interference with other components.

Radio Interference with Device Components:

- Interference in the end device is typically sourced from circuits such as CPU, memory chips, video circuits and other components generating high frequency noise which has the potential to couple into the radio through the antenna or other conducted paths. Such interference affects the overall wireless performance and User experience.
- The Embedded Device design must consider the integration of a 3G/4G radio module and minimise interference between the host system components and the 3G/4G embedded module and associated antenna subsystem.

Coexistence with other wireless technologies

- Attention needs to be paid to coexistence with other wireless technologies likely to be resident in the same unit. There should also be no interference between the 3G/4G radio interface and other radio interfaces present in the product.

6.5. Device Identification Codes

The IMEI ranges of the wireless modules embedded into the end product must to be submitted to Telstra on a regular basis (for Telstra Certified Devices). Submission details are provided upon initiating certification process.

- IMEI – International Mobile Station Equipment Identity – is a unique identity code used by network operators to distinguish between devices on our network.
- TAC (Type Allocation Code) refers first eight digits of the 15 digit IMEI code that details the manufacturer and model of a device.
- Telstra prefers that devices integrating modules to have a different TAC code to the embedded module - this allows easier management of the device group on the network. If this is not done then Telstra prefers a separate IMEI series within the TAC range to allow devices to be readily identified.

6.6. eSIM

An embedded SIM (eSIM) is a hardware secure element which holds the subscription profile of a mobile network operator that can be embedded/soldered into a device. eSIMs are reprogrammable, enabling remote provisioning and management of services over the air.

eSIMs are supported by the Telstra Network. Contact Telstra by email at TelstraWirelessM2MHardware@team.telstra.com to discuss deployment and use of eSIMs in your device.

6.7. Ruggedness

- Ensure device is appropriate hardened / rugged against the elements for remote field deployment as appropriate.
- Ensure device has sufficient protection to prevent theft of UICC (SIM). Device should have a sealable, tamper proof enclosure

7. APPLICATION DEVELOPMENT TEHCNIQUES

7.1. Best Practices for Development

Best practices that Telstra recommends when developing applications for use on our network are that apps should be designed to:

- Have interoperability / compatibility with the Telstra Network
- Minimize unnecessary data transfer
- Optimize necessary data transfer
- Be resilient to changing network conditions
- Be responsive
- Be secure
- Comply with industry and regulatory requirements
- Be serviceable
- Conserve power

A series of techniques are described in the following sections, referencing industry standards and guidelines that can assist developers in implementing best practices outlined above.

7.2. Fundamental Methods

7.2.1. Description

The GSMA Developer Guidelines outline a number of techniques to optimise performance of smart phone applications with mobile connectivity, which are equally applicable to IoT/M2M applications. The guidelines recommend use of the below listed methods for developing the ideal mobile application, touching upon many of the best practices listed in Section 7.1.

7.2.2. Methods

Asynchrony

To maximize user satisfaction, apps should be designed to be responsive which can be achieved using asynchronous logic for the main code block of an app.

- Make use of separate parallel threads for independent network requests
- The main application thread handling the user interface should not be blocked by outstanding responses to network requests.
- Progressively load and present network response/data as it arrives to the user. Do not wait for all responses to return successfully before providing an update to the user.

Connection Loss and Error Handling

- **Request types**: Categorise network requests as user initiated (primary), non-user/system initiated and secondary (spawning from primary requests) to determine appropriate actions in event of network issues
- **Cancellation**: Allow users ability to cancel primary requests. Cancellation of primary requests should result in cancellation of secondary requests.

- **Error handling**: Make use of notifications upon failure of primary requests. After attempting some limited number of retries, suspend the request and present the option to resume the request manually. See Section 7.3 for guidance on appropriate use of retry mechanisms.
- **Download resumption**: Divide large download files into chunks to make use of download resumption in event of network errors. This is an important mechanism to recover from interrupted file transfers rather than simply trying to download the entire file again.

Efficient Traffic Usage

- **Caching**: Keep a copy of the portion of data that has already been downloaded, in case it is needed again. Caching can reduce the need to reload images, web pages, style sheets, etc. which results in fewer data transfers, reducing network signalling and make apps appear faster and more responsive.
- **Cloud based transformations**: Avoid aggregation and processing of data from multiple data sources on the mobile application client. Instead perform these operations on an application server and expose its functionality as a web service via APIs to minimise the number of network connections and data transfer to the client.
- **Media transcoding**: Content optimization can minimize data usage and reduce download times. Developers should utilize the OS platform APIs/User Agent information to determine the device capabilities with regard to screen display resolution and streaming capabilities, and serve media accordingly. The lowest resolution/frame rate/codec rate that gives a good user experience should be used. Application Server should also have media content encoded in a variety of bit rates and the app should choose the media rate that suits the radio network being used.
- **Presence**: To minimise unnecessary traffic from presence based services information on presence or availability of users should be bundled before being published instead of sending/requesting an update per user separately. Make use of partial publication to only update information which has changed since the last state.
- **Email**: Consider imposing maximum attachment and message size limits to reduce the amount of data transfer. Provide users with a choice to download large attachments/messages instead of doing so automatically. Make use of Push notifications from server to device instead of polling to update messages.
- **Push notification**: Many applications attempt to deliver real time news, notifications and other data to devices by periodically polling the network which is wasteful if there is no new information on the server and causes unnecessary network signalling and drain on device battery. Instead push data to the device when there is actually new and relevant information available.
- **Compression**: Data compression where possible can be used to minimize data transferred over the network and reduce costs for the user. Applications that are text based and use HTTP protocols such as news aggregators lend themselves well to compression techniques, which can reduce text data size by 80%.
- **Data batching**: It takes time, power and network signalling to switch between device RRC (Radio Resource Control) states. When a device switches from an idle to dedicated channel to send data it consumes 60-100 times the amount of power it does in the idle state. By batching data, we save on these state changes, reduce battery drain and network signalling (which is beneficial for other network users). Batching applies on both the uplink (device to network) and downlink (network to device) sides.

Background Mode

Background mode refers to when the user interface of the application is not visible to the user and the app is not actively being used, in a multitasking environment. Once an app is placed in the background, the user might reasonably expect the app is not doing any data transfers. This however is not always the case.

Avoid network chattiness for your app in background mode – unless it is very clear to the user that the app will still work in background mode.

Cease relevant activity of app when it is placed in background e.g. stop video/ audio streaming , network connection and so on until app is brought into the foreground but continue other activity that might be required e.g. keep track of videos user has watched etc.

Give user an option in the app settings to stop app data transfer in background use. Some apps continue to run when in the background including transferring data which might not be what the user wants.

Application Scaling

Scale the application's network activity and behaviour depending on the available power reserves, and network conditions to extend battery life and provide a good user experience.

App should be developed to scale functionality according to the capabilities of the network it is connected to. More data intensive functions of the app should be limited to meet available network throughput. For instance:

- When on 3G, consider restricting video streaming functionality as the network speeds will not be sufficient to support at high quality.
- Scale back the codec rate of video delivered when connected to lower speeds, but offer higher quality video streaming and image resolution when on the faster 4G network or Wi-Fi.

In order to extend device battery life, app activity should be ratcheted down as battery charge declines. Some possible app activities that could be scaled are:

- Reduce periodicity / frequency of app updates or polls as battery declines.
- Reduce retry algorithms in low battery situations so as to not hasten a flat battery
- Do not allow certain activities without user warning and acceptance
    - E.g. once battery reaches a certain limit, do not allow certain activities such as uploads / downloads of large files, streaming, GPS activation etc. Inform user that battery is low and connection to a charger is recommended to continue activity (allow user to override however).

Device battery life can be extended by deferring non time critical uploads/downloads until charging. Consider settings options for your app to only upload / download large files such as captured photos and videos when charging.

7.2.3. References

Smarter Apps for Smarter Phones v4.0: http://gsmaterminals.github.io/Developer-Guidelines-Public

Implementation of these methods often requires use of platform specific APIs. Guides for Android and iOS platforms are linked below:

Google Android:	Connection management https://developer.android.com/guide/topics/connectivity Push notification https://firebase.google.com/docs/cloud-messaging/ Battery State http://developer.android.com/reference/android/os/BatteryManager.html
Apple iOS:	Connection management https://developer.apple.com/library/archive/documentation/NetworkingInternetWeb/Conceptual/NetworkingOverview/Introduction/Introduction.html Push notification https://developer.apple.com/library/archive/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple_ref/doc/uid/TP40008194-CH8-SW1 Battery State http://developer.apple.com/library/ios/#documentation/uikit/reference/UIDevice_Class/Reference/UIDevice.html

7.3. Network Connection Efficiency

7.3.1. Description

Mobile applications should consider the frequency and timing of their network connection establishment. Mobile devices need to transition through different devices states of increasing power draw and initiate several signalling events in order to setup a connection with the mobile network for transfer of application data. Misbehaving applications can be disruptive to the network and pose a significant drain on device battery.

7.3.2. Methods

Use Conservative Retry Algorithms

Conservative retry algorithms are required to prevent apps from continually trying to upload or download content in the event of failures such as server issues, timed out activities or slow network speeds. This is to prevent rapid battery drain and reduce harm to other network users.

The idea is to limit aggressive retry attempts and to have a sensible back off timer and a finite retry algorithm. A sensible retry algorithm will have a randomized back off time (before retry), with increasing time between retries, and a finite number of retries before indicating failure to the user/application.

In an IoT/M2M application, conservative retry algorithms are critical. Consider the utility IoT/M2M monitoring case where thousands of IoT/M2M devices might concurrently try to reconnect to their server after a power outage to upload their measurements. Without a well designed retry algorithm this can cause network access congestion, affecting not only the M2M specific application but other network users.

Another consideration for IoT/M2M applications is the effects an unsuitable retry algorithm could have on the battery life of the device. The importance of getting the device back online as soon as possible needs to be weighed up against the power requirements of having a more aggressive retry algorithm.

Avoid Synchronised Access to the Network

Smartphones and tablet devices are often synchronized to a common central clock. If multiple apps use absolute times to synchronize simultaneously to perform actions like fetch email or news updates, they may cause blocking at the local cell level or excess load across the entire network.

IoT/M2M devices in particular typically exist in large numbers with many sharing common network access points in the form of the local mobile base station. If all these devices were to attempt to signal the network simultaneously it would create a lot of congestion.

To avoid synchronized app access to the network, activities shouldn't be scheduled for the exact same absolute time across large applications. In an IoT/M2M scenario, we would not want all meter readings for a utility to occur at exactly the same point in time across an entire city. Similarly we would not want an email client set to check email at exactly 8am each morning.

Developers should consider spreading /randomizing device access by random offsets that are relative to the nature of the activity. For instance for a periodic IoT/M2M activity that requires a small transfer to occur hourly, spread the accesses for the devices randomly across the hour.

For something large such as a large software/firmware update consider randomizing device updates over a longer period such as week or a month and consider doing the data transfer at an off peak time such as in the middle of the night between Midnight and 6am.

For IoT/M2M solutions make sure you consider the case where a power outage (for devices that use mains power with no battery back-up) results in all the devices powering back up at the same time. These devices should not all try to reconnect to the network at the same time. Stagger the network activity of all the IoT/M2M devices so as not to cause network congestion.

Note Telstra's terms and conditions also mandate conditions around synchronized access to Telstra's network for multiple IoT/M2M modem devices and contains the following clause "If your Wireless M2M application employs more than 50,000 modem devices, you must provide a facility to control data transmission intervals in real time. We may require you to increase data transmission intervals during periods of network congestion".

7.3.3. References

See Section 7 – Connection Efficient Requirements from GSMA IoT Device Connection Efficiency Guidelines v5.0:

https://www.gsma.com/iot/wp-content/uploads/2018/06/GSMA-IoT-Device-Connection-Efficiency-Guidelines_TS.34_v5.0.pdf

Refer *"Use of multiple modem devices"* in Telstra's *Our Customer Terms*.

https://www.telstra.com.au/content/dam/tcom/personal/consumer-advice/pdf/business-b/dataservices-m2m.pdf

See Appendix B – WIRELESS TECHNOLOGY INFORMATION for more details on the relationship between device state transitions, network signalling events and power consumption.

7.4. Firmware over the Air Updates

7.4.1. Description

Firmware over the Air (FOTA) upgrade is the ability of a device to have its firmware/operating system and RF chipset firmware upgraded using the cellular network ("over the air").

Note FOTA can be achieved by proprietary methods or standardized methods such as described by OMA (Open Mobile Alliance) Specifications body (http://openmobilealliance.org/) in which case it is using the OMA-DM (OMA Device Management) standard on FUMO (Firmware Update Management Object). Telstra has no preference for method used. Telstra expects the vendor/OEM or integrator to host the FOTA server for their devices.

Device vendors should keep up with latest OS and/or module firmware releases, rolling out updates as they become available to ensure new security vulnerabilities are addressed and known issues are minimised.

Note that Telstra reserves the right to insist on a firmware upgrade using this capability at any time should we find device issues causing network harm/other user harmful impacts.

For IoT/M2M devices and applications

FOTA capability is extremely important for IoT/M2M devices given the:

- Large number of devices in remote or hard to access locations
- Longer lifecycle / lifespan of these devices compared with smartphones. FOTA is important because if any bugs are found in the device while in the field, or any future incompatibilities are found between the device and our radio network, these can be remotely rectified by the vendor/manufacture.

For IoT/M2M developers seeking Telstra endorsement of their IoT/M2M device or solution, it is mandatory that there is a mechanism to update the firmware of both the cellular modem and the software of the integrated device remotely. In some rare cases exceptions may be made and other mechanisms for upgrade may be allowed e.g. fixed internet connectivity, Wi-Fi, cabled connection.

7.5. Design with IPv6 Transition in Mind

7.5.1. IPv6 Requirements

Telstra is committed to implementing the IP version 6 protocol (IPv6) for communication with mobile devices connected to its network. Telstra is progressively enabling APNs for IPv6 use.

Telstra.internet APN has been enabled for IPv6 Dual Stack usage.

Telstra.wap APN has been enabled for IPv6 Single Stack usage for select tablets and handheld devices.

Once IPv6 single stack capability is enabled in the network, new consumer mobile devices (e.g. handsets, smartphones and tablets) that support 464XLAT will be connected using IPv6 only (or "IPv6 single stack", with the "IPv6" Packet Data Protocol or PDP type).

Devices without 464XLAT support will be connected either using simultaneous IPv4 and IPv6 connections ("dual stack") with the IPv4v6 PDP type, or else with IPv4 only (as now).

Telstra-homed devices roaming on other networks will only use IPv4 for connections, until such time that there is general support for IPv6 amongst global mobile carriers.

IPv6 Support – Smartphones & Tablets

Telstra expects handsets and tablets to support IPv6 with 464XLAT capability. Applications should be designed to use IPv6 as soon as it is available in the network. Current Android version supports XLAT.

IPv6 Support – IoT/M2M

IoT/M2M devices have the least IPv6 support as they tend to be the most cost sensitive devices and therefore are often produced using older chipsets with less feature support. Importantly IoT/M2M devices have quite long life cycles compared to Smartphones. So it is even more vital that IPv6 is considered when developing IoT/M2M applications. Developers should choose devices supporting IPv6 and ideally develop IPv6 compatible applications.

IoT/M2M devices using a custom APN may be enabled to use IPv6 only (single stack) on a case by case basis. Telstra can be contacted by email at TelstraWirelessM2MHardware@team.telstra.com to discuss this if required. Telstra expects future large-scale deployments of IoT/M2M devices will be configured to use IPv6 only (single stack).

Telstra will be requiring module manufacturers to support IPv6 going forward.

IPv6 Support – Wireless Broadband Devices

Wireless broadband devices (USB Dongles, Wi-Fi hotspots, Gateways) will be required to support dual stack IPv4/IPv6 connections using the IPv4v6 PDP type.

7.5.2. Methods

Some basic app guidelines for IPv6 are:

- Don't hardwire IPv4 addresses in to your app / app code – use variables to represent IP addresses.
- Ensure when coding to use a variable for IP addresses that can hold an IPv6 address
- Ensure apps are both IPv4 and IPv6 compliant
- For IoT/M2M solution developers - use IPv6 capable modules in your application and particularly Dual Stack IPv4v6 capable embedded modules in your device once they are readily available (not widely available currently – but at some point in the future). IPv6 capability will help future proof your device/application – increase its longevity and increase its security (if new security features of IPv6 are utilized).
- Developers using servers as part of their application should ensure that their servers are dual-stacked or IPv6 enabled. The application should be designed to provide IPv4 End to End or IPv6 End to End, with IPv6 E2E being preferred.

7.5.3. References

IPv6 General Reference

Refer to Appendix C for an explanation of IPv6 terminology
https://www.internetsociety.org/deploy360/ipv6/faq/

XLAT464 References

http://tools.ietf.org/html/rfc6877
https://sites.google.com/site/tmoipv6/464xlat

Refer to RFC7849: An IPv6 Profile for 3GPP Mobile Devices for recommendations on connecting to IPv6 networks while also ensuring IPv4 service continuity https://tools.ietf.org/html/rfc7849

7.6. Follow Security Guidelines

7.6.1. Description

Developers need to consider security and privacy aspects of their application in order to protect their users and their data.

Developers need to consider the following when developing applications

- Security of users sensitive information
- Fraud Prevention
- Provide an OTA (over the air) software/firmware update mechanism – so any identified security issues can be quickly patched
- Certificate management

7.6.2. Methods

Some methods developers can employ to ensure the security of their app and its data:

General Guidelines:

- Use the respective OS platform's app store update mechanism to address any app security issues ASAP
- Do not store or send user passwords or any other sensitive information in unencrypted text
- Use secure protocols such as SSL/TLS for transmitting any sensitive information over the network
- Enforce higher security password requirements on the user. e.g. A mixture of upper & lower case, alpha numeric & special symbols, and lengths > 6 characters
- Ensure that no sensitive information is stored in the app log files
- Test the app to ensure that passwords / authentication cannot be bypassed
- Minimize app platform permissions to only the absolute minimum necessary so as to minimize vulnerabilities and increase user confidence
- Developers should use well known standardised security libraries / third party software APIs that provide security/encryption functions that have been well tested in the market (and hardened/patched against known vulnerabilities)

- Note many of the platform OS security protections can be circumvented by 'jail breaking' or 'rooting' the device – so ensure that app code uses its own security mechanisms beyond those provided by the OS platform.
- Developers should only distribute their app via the official OS platform's app store and not make the app package available for distribution elsewhere. Malicious code can be inserted into standalone versions and redistributed versions of the app can leave users vulnerable to identity theft and various forms of malware.

IoT/M2M Application Specific Guidelines

Security cannot be trivialised – especially considering some of the main applications of IoT/M2M. The impact of security and hacking breaches can be extremely serious.

IoT/M2M devices are key to emerging industries such as smart grids and health monitoring. Needless to say security (and privacy) breaches in these cases could have life threatening and wide spread community impact.

Security is also needed for Fraud prevention – given that these devices and their SIMS may be relatively accessible in high numbers.

Some specific security measures for IoT/M2M devices include:

- Firmware update capability (OTA) to allow device to be quickly patched should any security issues / vulnerabilities come to light.
- Ensure the physical security of the SIM in the device. For instance to avoid the oft-cited scenario where utility meters sim cards are stolen and used for data/call theft.
- Use IPv6 - due to its enhanced security features
- Utilize vendors FOTA to ensure you have the latest firmware for your device
- Review module and OS development platform security guidelines
- Ensure device has sufficient password protection / user authentication procedures to prevent against hacker access
- The physical security of devices when installing. Consider alarming device back to central server e.g. alarm if enclosure is opened
- Utilize external consulting/testing expertise against hacking/intrusion for critical IoT/M2M applications in utility and health monitoring areas.
- Consider hiding SSID for Wi-Fi connected devices. No need to broadcast.

7.6.3. References

OS Platform Security Guidelines

Each of the major mobile OS platforms has its own security guidelines for developers. These are a very good reference for developers.

Google Android:	http://developer.android.com/training/articles/security-tips.html
Apple iOS:	https://developer.apple.com/library/mac/documentation/Security/Conceptual/SecureCodingGuide/Introduction.html

Android Developer Security Tips (whilst specific to Android contains principals that are applicable to all platforms): http://developer.android.com/training/articles/security-tips.html

GSMA IoT Security Guidelines - Covers security considerations for IoT endpoints (devices), network elements and services ecosystem infrastructure: https://www.gsma.com/iot/iot-security/iot-security-guidelines/

7.7. Follow Privacy Guidelines

7.7.1. Description

Telstra respects user's privacy and it is important that developers ensure users privacy. There are legal penalties for not following the applicable laws.

7.7.2. Methods

- Ensure that no user information is sent to third parties without the user being clearly informed and opting in to the service (they must be clearly informed of where the data is going and how it will be used)
- GPS should not be used without the users consent to track / record user location
- Ensure security guidelines above are adhered to as these will help safeguard the users privacy
- Ensure your app complies with Australian Privacy Laws.

7.7.3. References

Australian Privacy Laws: https://www.oaic.gov.au/privacy/the-privacy-act/

http://www.futureofprivacy.org/wp-content/uploads/Best-Practices-for-Mobile-App-Developers_Final.pdf

7.8. Conclusion

In conclusion, developers should consider the guidelines mentioned throughout this section.

One thing to bear in mind is that the general principles apply to both data transfers and signalling that occurs due to network accesses and detaches.

There should be randomization, sensible back-off algorithms and non-infinite retries for all types of network use - both data transfer and network accesses. Appendix D – Network cause codes and device behaviour is highly relevant to this and details some common Network Cause codes and how devices should behave upon receiving them.

8. APPLICATION TESTING

8.1. Importance of Testing:

One of the most important aspects of app development is testing.

Testing prior to deployment has obvious advantages -

- Less chance of significant errors being found by users
- Ability to tune app performance
- Ability to observe network interoperability performance
- Ability to tune device battery performance
- Ability to observe any unintended behaviour and rectify

8.2. General Testing Guidelines

- Field testing of the app should be performed rather than just using a simulation tool
- App should be tested on all networks it could end up on i.e. 3G, 4G and Wi-Fi. Performance differences should be noted and code optimization done as a result
- App should be tested in areas with poor coverage / network connectivity to see how robust / resilient the app is to poor network conditions and connectivity and whether further app optimization is required to account for these issues.
- App should be tested in peak times (for data apps typically around 9pm on a weekday) to see how it performs under busier network conditions where there may be additional delays in responses from network
- App should be tested against any user configurable settings that are possible i.e. if user can set times for push to be disabled – check that these actually are disabled at the set time, or if uploading of photos is only permitted on Wi-Fi, make sure that this occurs.
- Monitor battery usage with the app (most operating systems provide battery monitoring tools and app stores have battery monitoring apps)
- Monitor data usage of the app. Check that it is consistent with expectations
- Ensure app server responds as expected in all the different test cases and network conditions as possible
- Try as many different combinations of usage cases as possible to detect any unintended UI behaviour
- Try to test using a variety of device models to ensure your app caters properly for different screen types, resolutions, device types (tablets vs. handsets)
- Consider using an external test house that has a large selection of devices, or use crowdsourcing web sites to get beta testers using a variety of devices
- Use User Agent switchers on desktop browsers to see how mobile web sites will be rendered on different devices.
- Check app behaviour with no network connectivity (offline). Does it provide access to functionality that doesn't require network connection or does it hang?
- Check that app performs as expected when in airplane mode or without a sim (i.e. no network connectivity)
- Test performance after network connectivity is restored after losing it

- Check that app performs as expected when there is a cabled connection to computer (tethered)
- Check interaction with peripherals such as SD card, Bluetooth accessories.
- Check that app performs as expected with above peripherals on/off/connected/not connected as appropriate.
- Check that app gives user appropriate indication of network connectivity, error conditions and ensure that there is a non-blocking UI. e.g. If no network connectivity the device doesn't flash up an error message and get stuck but rather allows the user to continue with activities within the app that don't require network activity.
- Ensure Security is tested. e.g. user authentication works as expected
- Does the app correctly handle interruptions e.g. from incoming calls.
- CPU usage of app is not excessive (most operating systems provide CPU monitoring tools)
- That the app performs as expected when the device is multitasking with other apps or when the app is background mode.

8.3. IoT/M2M Specific Testing Guidelines

Given the remote locations and rugged environments that IoT/M2M solutions can exist in testing must be all the more rigorous.

In addition to the testing described above the following additional testing is required for IoT/M2M:

- Test that FOTA solution works – test that both the application software can be updated and verify that firmware upgrade solution will work.
- Test that device is sufficiently rugged for planned deployment/usage. Is the device sufficiently hardened for the expected environment?
  - Test for heat, vibration, moisture, UV exposure, and generally adverse weather
- Test for failure conditions e.g. for power metering application, what happens if there is a power failure?
- Test that device over the air diagnostics work
- Verify that network reacquisition and retry algorithms function in a finite and non-aggressive way
- Test IPv6 functionality

8.4. References

http://www.appqualityalliance.org/online_testing_tool_and_best_practice_update

http://www.appqualityalliance.org/resources

9. APPENDIX A – TELSTRA WIRELESS AND IOT/M2M RELATED PRODUCT INFORMATION

9.1. Telstra Mobility Partners

Telstra has dedicated IoT/M2M product and solution teams to assist developers with integrating IoT/M2M solutions within our network. Email TelstraWirelessM2MHardware@team.telstra.com for assistance.

9.2. Telstra IoT Offerings

https://www.telstra.com.au/business-enterprise/products/internet-of-things

9.3. Telstra IoT Platform

https://www.telstra.com.au/business-enterprise/products/internet-of-things/capabilities/cumulocity

9.4. Telstra IoT Connection Management

https://www.telstra.com.au/business-enterprise/products/internet-of-things/capabilities/connection-management-platform

9.5. Telstra Mobile Assets and Workforce Enterprise Solutions

https://www.telstra.com.au/business-enterprise/solutions/mobility-solutions

9.6. Telstra Wireless Managed Data Networks – Wireless WAN

https://www.telstra.com.au/business-enterprise/download/document/business-mdn-wireless-wan.pdf

9.7. Telstra IP VPN information

https://www.telstra.com.au/business-enterprise/solutions/network-services/connectivity/vpn-service

9.8. Telstra Enterprise Support Contacts

https://enterprise-support.telstra.com.au/t5/tkb/communitypage

9.9. Telstra Mobile Phones

https://telstra.com.au/mobile-phones

9.10.Telstra Mobile Coverage

https://www.telstra.com.au/coverage-networks/our-coverage

10. APPENDIX B – WIRELESS TECHNOLOGY INFORMATION

10.1. RRC State Diagram

Network signalling is related largely to the RRC (Radio Resource Control) state transitions of the wireless device. It is important not to have unnecessary network signalling as this decreases the networks performance in terms of responsiveness and the amount of traffic it can handle for all users. It can therefore make an app appear slow or unresponsive and excessive state changes can quickly drain a devices battery.

There are some app behaviours that cause unnecessary RRC state changes leading to too much signalling, which should be avoided, including –

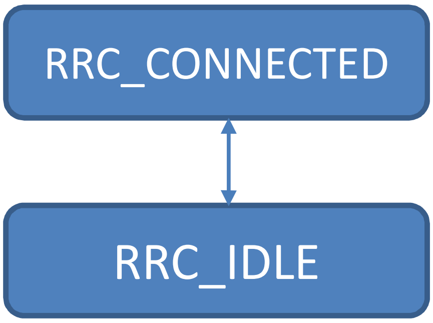
- Having heart beats/pings to maintain always on activity
- Poor Network reattachment algorithms.
- Constant polling applications

RRC State Machine

The RRC state machine describes how a wireless device (4G) connects to the radio network in a variety of different states – which have different levels of connectivity, power consumption and throughputs.

This discussion will concentrate on the 4G network as this network has many advantages for users, developers and Telstra as an operator, over the older 3G technology and provides a better user experience.

LTE RRC STATE DIAGRAM



As the device moves up the RRC states stack it consumes increasing amounts of energy, and the throughput available from the device increases. Note that these state changes require network signalling. There are also network controlled timers that control the minimum time before a state change can occur. It takes time to change state, and for small blocks of data the time to change state may actually exceed the time to transmit the required data. Therefore it is sensible to batch up small amounts of data into larger blocks.

LTE or 4G has two main states – RRC_CONNECTED and RRC_IDLE. An app that is in the connected state might typically consume a few watts whilst in IDLE state it will only consume tens of milli-Watts. The RRC_CONNECTED state is the state for data transfer between the network and mobile device. LTE is designed to allow the device to move between these two states very quickly especially when moving from IDLE to CONNECTED. Once in CONNECTED mode a network timer will again determine how quickly the device moves back to IDLE. So again to maximise device battery life it is important to reduce the amount of small data transfers to the network. The recommendation to buffer, collect and forward intermittently.

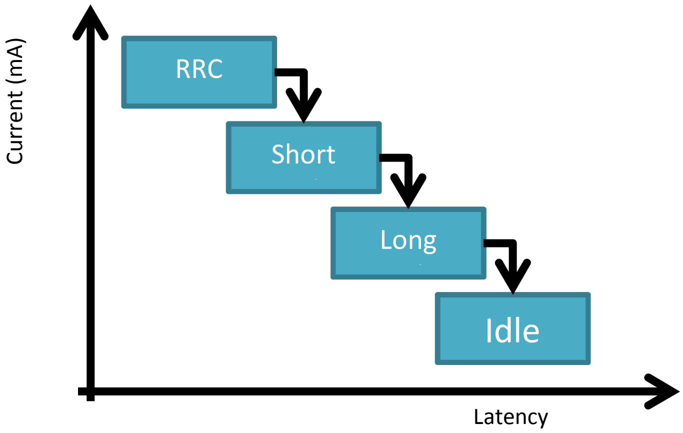
CDRX

CDRX stands for Connected Discontinuous Reception. It is an LTE device feature that allows the device to have micro sleeps and is controlled by the network. It helps to reduce network signalling load which benefits all users of the network.

It is a feature that reduces device battery consumption while minimizing impact on latency. In Telstra's network CDRX is only invoked after 100ms of inactivity.

The point to note is that the UE is still RRC connected while in CDRX sleep states.

The diagram below shows CDRX states for 4G



The point of all the above discussion is that it takes time, power and network signalling to switch between these RRC states and so it makes sense to minimize unnecessary state changes

10.2.Data Session Setup

Some brief information regarding data session on 4G/LTE.

LTE always has IP connectivity which is established by a **Default** Evolved Packet System (**EPS) bearer** by the Network Attachment procedure. This procedure attaches to the network and sets up a Default EPS bearer – and allocates IP address to the device.

When a device attaches to the LTE network for the first time it will be assigned a default bearer, which stays all the time and provides the always on IP connectivity. It has a nominal QOS.

If a specific QOS is required then this can be achieved by the network setting up a **Dedicated** Bearer. Currently Telstra does not yet have a specific QoS implementation for different applications / M2M so dedicated bearers should not be used for any purpose. UE initiated QoS is not supported and should not be attempted.

If an application requires a non-standard APN, then a new default bearer to the APN is established.

Developers should discuss any custom APN needs with Telstra via email address TelstraWirelessM2MHardware@team.telstra.com

A default bearer remains as long as the UE is attached to the LTE network. A UE can have additional default bearers. Each default bearer has its own IP address.

LTE dedicated bearers must be paired with a default bearer and they use the same IP address as the default bearer.

An LTE device can have up to 18 bearers in total – but one must be a default bearer. A default bearer is needed per APN (and another IP address).

IoT/M2M developers need to consider network connection setup procedure for 4G when developing their solution e.g. in the case of prompt for password applications on 4G where the device may connect to the network and fail if the correct password has not been previously entered.

Developers should consult with their vendor's documentation on relevant high level software commands or low level AT commands to perform these procedures as needed.

10.3.References

Refer 3GPP TS 36.331 for more information on RRC states of 4G devices:
https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2440

Refer 3GPP TS 24.301 for more information on EPS session management procedures:
https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1072

11. APPENDIX C – IPV6 DESCRIPTION AND TERMINOLOGY

The drivers for the use of IPv6 and the description of IPv4 exhaustion have been well documented on the Internet, and will be assumed to be understood by device application developers – see the Wikipedia article http://en.wikipedia.org/wiki/IPv6 for an introduction to IPv6 if required.

IPv4 and IPv6 are distinct protocols that do not natively interoperate, but it can be assumed that both protocols will need to coexist in the Internet for many years to come. This means that any device will need to communicate with other devices that may themselves either speak only IPv6 or IPv4, either by itself natively speaking both protocols ("dual stack" configuration) or through a protocol translator or tunnel in the network or the device (e.g. NAT64, 464XLAT, etc.).

Single Stack

A single stack network or device as the name implies only supports a single IP protocol type. Single stack is often denoted simply as SS.

There are two possibilities:

1. IPv4 SS network, supporting only IPv4 traffic as most existing wireless networks are today
2. IPv6 SS network, supporting only IPv6 traffic

Dual Stack

A dual stack network is a network whose nodes are capable of processing IPv4 and IPv6 traffic simultaneously. It thus facilitates the transition to IPv6 while many devices and internet sites are still IPv4 by letting the two protocols co-exist.

A dual stack UE (user equipment aka mobile device), supports the following PDP types IPv4 (single stack), IPv6 (single stack) and IPv4v6 (dual stack or "DS" – that is an IPv4 and an IPv6 connection simultaneously).

A dual stack UE will request a dual stack bearer (IPv4v6) and the network will allocate the appropriate bearer which may be either IPv4 single stack, IPv6 single stack, or dual stack (both an IPv4 and IPv6 connection).

464XLAT

For handsets & tablets that are IPv6 single stack capable there will still be a need for many years to co-exist with the IPv4 ecosystem. E.g. to reach IPv4 sites, or for apps that are not yet IPv6 compatible (i.e. contain IPv4 literal addresses within their code).

RFC6877 464XLAT provides a solution to allow IPV4 services & applications to work over an IPV6 single stack network. The 464XLAT solution requires at a minimum a NAT64 in the network along with 464XLAT daemon/code running on the device.

464XLAT refers to architecture for both network and device to allow this to work and this is described in RFC6877. Refer http://tools.ietf.org/html/rfc6877. 464XLAT was first proposed by T-Mobile (Cameron Byrne) in partnership with NEC and JPIX in this RFC.

The 464XLAT code required by the device is open source code and is available to be used in any operating system.

12. APPENDIX D – NETWORK CAUSE CODES AND DEVICE BEHAVIOUR

The network has a variety of cause codes that it can send to the device in response to device requests that indicate a reason for failure of the request.

The device should pay attention to these cause codes and behave accordingly.

In a few cases the industry (3GPP) specifications and network timers specify the retry algorithm's behaviour but not always completely. Thus in these cases and where not specified at all, the behaviour is left to the device manufacturer/integrator. As discussed earlier, if retries are required, they should not be aggressive and infinite in nature.

Some cause codes indicate trouble with a user's service subscription and retries are pointless (once issue is confirmed as not a one off) – in this case the user/developer needs to confirm their service subscription status with Telstra. For 3G these cause codes are described in 3GPP specification 24.008 annexes.

Refer http://www.3gpp.org/DynaReport/24301.htm
There are three main categories of error codes. Those related to Mobility Management (MM), Call Control (mainly applicable to voice calling) and Data Session Management (SM).

Mobility Management (MM) includes causes related to:

- MS Identification,
- Subscription Options
- Network Failures/Congestion/Authentication Failures
- The nature of request
- Invalid messages
- GMM = related specifically to data

Call Control (CC) causes are grouped into:

- Normal Class
- Resource unavailable class
- Service / Option not available
- Service Not implemented class
- Invalid Message
- Protocol error
- Interworking issues

GPRS/Data Session Management (SM) [ESM used for LTE] are divided in to subgroups of causes related to:

- Nature of request
- Invalid Messages

The table below summarizes some commonly seen codes, their meaning and suggested device behaviour.

Cause Code	Meaning	Scenario where it may occur	Proposed Device Behaviour or Developer Action Required
8	Operator Determined Barring	Service is barred	Don't retry - contact Telstra Support to ascertain why service is barred.
26	Insufficient resource	Network has insufficient resources e.g. Congested	Since network is congested wait and try again. The device shall not enter an endless retry mechanism. After each rejection, the device shall introduce a back off timer (recommended 12 minutes). We suggest doubling this back off timer after each rejected request and ultimately stop the requests after a period of time (recommended 2 days). Device must still comply with relevant 3GPP standards and obey applicable network timers Consider sending data during off peak times (midnight - 6am)
27	Missing or unknown APN	Incorrectly configured device settings for APN profile	Confirm with Telstra that you have correctly configured and are using the correct APN for the application After each rejection, the device shall introduce a back off timer (recommended 12 minutes). We suggest doubling this back off timer after each rejected request and ultimately stop the requests after a period of time (recommended 2 days).
28	Unknown PDP address or PDP type	3G specific analogous to 54 for LTE. Possibly due to incorrect internet destination configured in device	The device shall not enter an endless retry mechanism. After each rejection, the device shall introduce a back off timer (recommended 12 minutes). We suggest doubling this back off timer after each rejected request and ultimately stop the requests after a period of time (recommended 2 days). Device must still comply with relevant 3GPP standards and obey applicable network timers. Developer should check device configuration and internet settings Seek Telstra technical support if error persists
29	User Authentication Failed	Service or SIM error	If this occurs more than once, then contact Telstra Support to confirm service subscription is correct and to investigate the issue.

30	Activation rejected by GGSN, Serving GW or PDN GW	This error occurs if the device/application is requesting a service that is not supported by the network.	Again device shall not endlessly try - but rather stop and provide meaningful error to the user. Developer to investigate what service is being requested and confirm with Telstra whether the service is supported and if not what an equivalent alternative service would be suitable. Again device shall not endlessly try - but rather stop and provide meaningful error to the user
31	Activation rejected, unspecified	Possibly due to the requested service option not being subscribed to or other reason	The device shall not enter an endless retry mechanism. After each rejection, the device shall introduce a back off timer (recommended 12 minutes). We suggest doubling this back off timer after each rejected request and ultimately stop the requests after a period of time (recommended 2 days). Device must still comply with relevant 3GPP standards and obey applicable network timers. Contact Telstra for support
32	Service option not supported	Occurs when the network doesn't support the service option.	Again device shall not endlessly try - but rather stop and provide meaningful error to the user. Developer to investigate what service is being requested and confirm with Telstra whether the service is supported and if not what an equivalent alternative service would be suitable. Do not automatically retry. May occur in a roaming network.
33	Requested service option not subscribed	Occurs due to the requested service option not being subscribed to.	The device shall not enter an endless retry mechanism. The device will not retry unless power cycled or a device setting is altered. Contact Telstra for support

34	Service option temporarily out of order	Likely due to network fault	Given that this is due to a network issue that is temporary trying again is reasonable. However the device shall not enter an endless retry mechanism. After each rejection, the device shall introduce a back off timer (recommended 12 minutes). We suggest doubling this back off timer after each rejected request and ultimately stop the requests after a period of time (recommended 2 days). Device must still comply with relevant 3GPP standards and obey applicable network timers If problem persists contact Telstra for support.
38	Network Failure	Likely due to network outage	Given that this is due to a network failure it is important not to repeatedly retry as this will make it difficult for the network to recover. Suggest backing of for tens of minutes before retrying. As always the device shall not enter an endless retry mechanism. After each rejection, the device shall introduce a back off timer (recommended 12 minutes). We suggest doubling this back off timer after each rejected request and ultimately stop the requests after a period of time (recommended 2 days). Device must still comply with relevant 3GPP standards and obey applicable network timers If problem persists contact Telstra for support.
50	PDP type IPv4 only allowed	Will occur if device requests an IP protocol type (e.g. IPv4v6) that is not allowed by the network or user subscription e.g. if requests IPv6 bearer when they aren't supported	Device shall set up an IPv4 bearer and not request IPv6
51	PDP type IPv6 only allowed	Will occur if device requests an IP protocol type (e.g. IPv4v6) that is not allowed by the network or user subscription e.g. if requests IPv4 bearer on IPv6 Single Stack network	Device shall set up a IPv6 bearer and not request IPv4 bearer
52	Single address bearer allowed		If device requests an IPv4v6 PDP and network sets the PDP type to IPv6 (or IPv4) with cause code #52 (single address bearer allowed), the device shall use the allocated IP address from the network. The device can subsequently request another PDP context activate for the other bearer if it requires dual stack connectivity if the network does not support IPv4v6 on one bearer.

53	ESM Information not received		
54	PDN connection does not exist	LTE specific analogous to 28 for 3G. Possibly due to incorrect internet destination configured in device	Developer should check device configuration and internet settings

13. APPENDIX E – APN TIMEOUTS

The following timeout info applies to all APNs that undergo NAT (Network Address Translation).

Inactivity timeout for general traffic:

TCP	30 min
UDP	2 min
ICMP	4 sec
DNS	5 sec

These timeouts are subject to change.

Extranet services (that do not undergo NAT) have the same timeouts applied TCP/UDP/ICMP/DNS on the Stateful firewall.

14. GLOSSARY

Abbreviation / Term	Definition
3G	3rd Generation Wireless Network based on WCDMA technology
3GPP	3rd Generation Partnership Project
4G	4th Generation Wireless Network based on LTE technology standards
5G	5th Generation Wireless Network based on NR technology standards
API	Application Programming Interface
APN	Access Point Name
CA	Carrier Aggregation
FOTA	Firmware Over The Air
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
LTE	Long Term Evolution
MB	Megabyte
M2M	Machine to Machine
OS	Operating System
OMA	Open Mobile Alliance
OTA	Over The Air
PSM	Power Saving Mode
RRC	Radio Resource Control
SDK	Software Development Kits
SMS	Short Message Service
SSL	Secure Sockets Layer
T&Cs	Terms and Conditions
TLS	Transport Layer Security
WCDMA	Wideband Code Division Multiple Access
UI	User Interface