Best Practices for 3G and 4G App Development

Whitepaper

Revision **1.0**
Revision Date **7/31/2012**

Legal Disclaimer

This document and the information contained herein (collectively, the "**Information**") is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) ("**You**" and "**Your**") by AT&T, on behalf of itself and its affiliates ("**AT&T**") for informational purposes only. AT&T is providing the Information to You because AT&T believes the Information may be useful to You. The Information is provided to You solely on the basis that You will be responsible for making Your own assessments of the Information and are advised to verify all representations, statements and information before using or relying upon any of the Information. Although AT&T has exercised reasonable care in providing the Information to You, AT&T does not warrant the accuracy of the Information and is not responsible for any damages arising from Your use of or reliance upon the Information. You further understand and agree that AT&T in no way represents, and You in no way rely on a belief, that AT&T is providing the Information in accordance with any standard or service (routine, customary or otherwise) related to the consulting, services, hardware or software industries.

AT&T DOES NOT WARRANT THAT THE INFORMATION IS ERROR-FREE. AT&T IS PROVIDING THE INFORMATION TO YOU "AS IS" AND "WITH ALL FAULTS." AT&T DOES NOT WARRANT, BY VIRTUE OF THIS DOCUMENT, OR BY ANY COURSE OF PERFORMANCE, COURSE OF DEALING, USAGE OF TRADE OR ANY COLLATERAL DOCUMENT HEREUNDER OR OTHERWISE, AND HEREBY EXPRESSLY DISCLAIMS, ANY REPRESENTATION OR WARRANTY OF ANY KIND WITH RESPECT TO THE INFORMATION, INCLUDING, WITHOUT LIMITATION, ANY REPRESENTATION OR WARRANTY OF DESIGN, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, OR ANY REPRESENTATION OR WARRANTY THAT THE INFORMATION IS APPLICABLE TO OR INTEROPERABLE WITH ANY SYSTEM, DATA, HARDWARE OR SOFTWARE OF ANY KIND. AT&T DISCLAIMS AND IN NO EVENT SHALL BE LIABLE FOR ANY LOSSES OR DAMAGES OF ANY KIND, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF GOODWILL, COVER, TORTIOUS CONDUCT OR OTHER PECUNIARY LOSS, ARISING OUT OF OR IN ANY WAY RELATED TO THE PROVISION, NON-PROVISION, USE OR NON-USE OF THE INFORMATION, EVEN IF AT&T HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES OR DAMAGES.

Table of Contents

1. Overview

4G LTE (Long Term Evolution) is the latest, fastest and most secure version of four generations of mobile technologies: 1G, 2G, 3G and 4G. Prior to LTE, voice and data were carried over two channels: A circuit switched channel for voice and an IP-based packet core network for data. LTE is designed to provide a single, simplified, all-IP channel to carry both voice and data. LTE is being deployed worldwide to meet demands from the exponential growth of mobile data traffic and is one of the three largest drivers of mobile data consumption[1]:

- **Faster networks:** LTE, with higher speeds and significantly lower latencies, enables the innovation of mobile apps with richer services, for example, video sharing in high-definition on the uplink for video conferencing or playing latency-sensitive games.
- **Powerful devices:** Computing power is bringing desktop quality computing to mobile devices (smartphones and tablets). Declining price-points for increasingly more powerful devices are resulting in the increased usage of Internet enabled mobile devices. For example, over 56% of AT&T's customers are already using smartphones[2].
- **Mobile apps:** Apps are considered the major driver of mobile data consumption. Video apps alone (e.g., Hulu, Netflix) consume 66% of all mobile data traffic[3]. Apps on powerful, faster mobile networks and mobile devices bring opportunities to developers that understand both the device platforms and the mobile networks (2G, 3G and 4G) in which they operate. Developers will be able to build more efficient and powerful mobile apps when they understand how networks function. This efficiency in turn makes networks more valuable so customer adoption of apps accelerates.

This white paper emphasizes LTE technical considerations rather than legal or business development issues. It covers the following main areas to help developers build LTE efficient apps:

1. **Evolution:** A history of the evolution of mobile networks from 1G to 4G.
2. **Benefits:** The benefits of developing apps for LTE.
3. **Apps:** Examples of LTE specific apps.
4. **Challenges:** The LTE rollout is a gradual deployment to U.S. cities which introduces issues that developers must consider while the LTE deployment is ongoing.
5. **Best practices:** Core principles for developers that directly impact their users.

2. Scope

The focus of this whitepaper is to address the best practices for the four key areas of application performance that customers perceive the most:

- **Data cap:** A significantly faster network hasn't translated (yet) into larger monthly data caps. Users are still dealing with 3 – 5 GB data caps.
- **Battery-life:** LTE handsets result in a heavier drain on battery life than earlier generations. This heavier drain represents the biggest area of concern for customers.
- **Speed:** Bandwidth and latency are the two metrics that users mean when they say an app is "fast" or "slow."
- **Robustness:** Apps must be robust in handling handover to slower 3G networks. Unlike other carriers, AT&T's LTE/HSPA+ combination will have a less jarring speed drop than other carriers that downshift between an LTE/3G combination.

3. Evolution of LTE

The following chart shows the evolution of the capabilities of mobile networks from 1G to 4G LTE.

	1G	2G	2.5G	3G	4G HSPA+	4G LTE
Year(s)	1980s	1990s	2001	2008	2010	2011
Voice calls	✓	✓	✓	✓	✓	✓
Digital		✓	✓	✓	✓	✓
SMS		✓	✓	✓	✓	✓
Internet access		✓	✓	✓	✓	✓
Circuit Switched Data	✓	✓	✓	✓	✓	
IP-Based Packet Core			✓	✓	✓	✓
Industry Max Speed (Kbps)	-	9.6	40 – 50	3,800	6,000	61,000
Icon displayed	None	None	G or E	3G	4G or H+	4G LTE

1G: In the 1980s, a 1G cellular network handled only voice calls (no data) in analog waves at 9.6 Kbps. It had several limitations such as poor voice quality and a lack of privacy (third parties could eavesdrop on calls on analog networks) due to the absence of encryption.

2G: In the 1990s, second generation (2G) mobile phones and networks were introduced. Due to modulation (voice converted to digital signals in the phone and then analog for transmission), handsets overcame some 1G limitations such as a lack of privacy by adding encryption. Unlike 1G phones which only supported voice calls, 2G also supported data services such as SMS and WAP (Wireless Access Protocol – a lightweight protocol for transferring data on handsets). Both voice & data were transmitted using a single channel, CSD (Circuit Switched Data) technology.

2.5G: This was the interim technology to 3G. 2.5G was launched in two major flavors: GPRS and EDGE. The letter "G" would be displayed on a handset's screen next to the bars if it was on GPRS (General Packet Radio Service) and "E" if it was on the faster EDGE (Enhanced Data for GSM Evolution). EDGE is approximately 3x faster than GPRS and was also known as 2.75G. GPRS had download speeds of 30-40 Kbps.

3G: By early 2008, the third generation, 3G, was launched with GSM UMTS (WCDMA, HSPA, and HSDPA). Phones display "3G" on the screen when using a 3G network. With 3G, voice and data travel on two separate data paths: CSD for voice and a packet core network for data. As a result, AT&T customers using 3G became the first users nationwide, to be able to use voice and data simultaneously.

4G: The ITU (International Telecommunications Union) defines 4G as the ability to achieve a theoretical peak download of 100 Mbps while mobile and 1 Gbps while stationary. This UN entity also extended the classification to include slower 4G forerunners which offer significant improvements over 3G such as HSPA+ (High Speed Packet Access) and its faster successor LTE (Long Term Evolution). 4G LTE was rolled out by AT&T in September, 2011. Phones typically display "4G" or H+ when on HSPA+ and "4G LTE" when on LTE.

The next evolution after LTE is LTE Advanced (LTE-A) with theoretical download speeds of 1 Gbps and 200 Mbps upload speed. However, deployment dates have not been announced.

4. Benefits

4G LTE brings four key benefits to mobile devices:

a. Higher bandwidth

Bandwidth is the rate at which data can be transferred via physical media such as fiber, copper or in the case of smartphones, via radio waves. Depending on device and network conditions, LTE is actually 7 to 15x faster[4] than the earlier generation i.e., 4G HSPA+.

Note that LTE speeds shown in the following chart represent peak data rates under ideal, lightly loaded network conditions. Actual speeds under real-world conditions will be less than those at peak rates.

4G Type	Definition	Download (Mbps)	Upload (Mbps)
HSPA+	High Speed Packet Access	6	1.2

LTE	Long Term Evolution[5]	100	50

b. Lower latency

In LTE, the latency or lapsed time between a request and its response has been reduced to Wi-Fi levels of less than 50 milliseconds (ms). The combined effect of higher bandwidth and lower latency enables mobile developers to deliver better experiences. That includes the use of data intensive technologies such as HD video streaming, cloud-based speech recognition, augmented reality, and interactive gaming.

Round-trip Latencies for cellular and wired environments

Environment	Type	Latency (ms)
Cellular (AT&T)	4G HSPA+	100 - 200
Wi-Fi (phone)	802.11n	50
Cellular (AT&T)	4G LTE	40 - 49
Wired	Ethernet	25

Note: Latencies in this table are for round-trip requests and responses that occur entirely in the continental U.S.

c. Improved security

Security improvements in LTE include the use of larger 128 bit network authentication keys when data is sent on a secure channel. This allows the RAN (Radio Access Network) to provide more secure communications between network elements and terminal devices. IPSec is also provided for secure tunneled IP communication if it is supported by the device.

d. Prioritized QoS

LTE Quality of Service (QoS) introduces nine types of traffic with individual processing priority. This means that latency-sensitive traffic can be prioritized over latency tolerant traffic. For example, conversational voice and (real-time) video with a higher QoS will not be delayed by lower QoS traffic such as email and file-transfers. QoS prioritization also means that voice data can be prevented from over-consuming bandwidth reserved for video broadcasts. Quality of Service is defined by the carriers and can be customized at the subscriber level. Some of the nine types of LTE traffic are conversational voice, conversational video, streaming video, and real-time games[6].

The policy and charging rules function (PCRF) is one of two dedicated network elements that define, control, and enforce policies in addition to managing QoS resources[7]. Currently, PCRF is controlled by carriers. Some examples of how PCRF can be used by carriers are: Providing a data cap increase for certain traffic types if bills are paid on time, and providing unlimited data usage after a certain time, similar to how cell phone users have unlimited talk time on nights and weekends.

Currently, there are no APIs provided to access PCRF. In the future, if PCRF APIs are provided, developers will be able to build that functionality into their services. For example, such a service could notify the carrier with an instruction to zero-rate video traffic temporarily when a paid subscriber of Hulu Plus watches HD video, while drawing down the data cap at the regular rate for an unpaid user of Hulu Plus watching the same HD content.

5. LTE Enhanced Applications

The promise of an LTE network, with broadband type speed and latency, opens a world of opportunity for the types of applications that developers can build. The following are some examples of the kinds of applications that could be enhanced by LTE.

Video apps: High-definition (HD) extensions of existing 3G and 4G HSPA+ video and video centric apps are obvious candidates for LTE enhancement. For example, a user may want to do a live HD broadcast of an event, via a server, to the smart TVs and phones of their friends, or conduct a private one-to-one broadcast to their TV. They may even want to conduct live, one-to-many, HD broadcasts, via server, to broadcast to the public.

Augmented reality: A new trend in mobile apps is to overlay information on a live view of a physical real-world environment. An example of this would be a user trying to identify if there's a sale in the immediate vicinity of their location. The user would pan the camera of a smartphone or tablet across a scene and tags would appear on the live video view. The user would then be able to click on those tags and get rich data (audio, video and text) specific to that store, such as coupons, menus, reviews, historical factoids, or user generated content. Augmented reality works today on 3G. However, where 4G LTE becomes vital is the spectral efficiency – fatter pipe – which allows larger numbers of people to view bandwidth intensive video linked to tags.

Interactive games: There are over 135 million gamers in the U.S.[8], and with the rise of MMOs (Massively Multiplayer Online games), gamers can now play latency sensitive and bandwidth hungry games such as *World of Warcraft* that are streamed directly to their LTE devices. In fact, in 2012, game streamers such as OnLive and Gaikai already stream hardcore game titles such as *Need for Speed* directly to broadband devices such as laptops and smart TVs. This is being extended to LTE devices as well which, because of their reduced latency, provide a better gaming experience than prior generation mobile devices. Note that streaming apps, as explained in the challenges section, are battery and data hungry and should be tested and optimized with tools such as AT&T's Application Resource Optimizer[9] (ARO) to minimize power consumption.

M2M: Mobile networks used to be only for voice communication. But now, the trend is to the Internet of Things or machine to machine (M2M) communication. Smart meters, traffic lights, cars, household appliances, and other devices are able to communicate with each other.

M2M is used in 2G and 3G[10]. The benefit that LTE brings to M2M is increased spectral efficiency. Spectral efficiency in 3G UMTS for example, is 3 bits per second (bps)/Hz. LTE, however, increases that five-fold to 15 bps/Hz. This increased efficiency becomes critical when forecasts show that the number of connected devices is forecasted to double to 12 billion by 2020 [11]. Thus, carriers are able to offload M2M data traffic from congested 3G to LTE.

One example of M2M is AT&T's *Digital Life* (digitallifeservices.att.com) that provides a remote monitoring and automation platform for connecting a user's device to devices in their home. This platform makes it possible to create apps for web-based home automation, security, energy management, and healthcare services.

6. Challenges

Network changes don't happen overnight. It has taken almost ten years for each wireless network generation to evolve and be deployed. The deployment of LTE networks is well under way, but it is expected to coexist with 3G and even 2G networks for years to come.

With all the claims around LTE and its blazing speed, it is important to understand that there are many factors that may affect the perceived speed. Such factors include tower proximity, packet fragmentation, concurrency, and signal blocking. The best practices section in this white paper addresses how developers can leverage LTE best practices to manage these challenges, and it introduces a software tool to help developers visualize issues and evaluate how compliant their app is with these best practices.

a. Channel quality

In wireless networks *channel quality* is measured by the amount of successful packets delivered. Channel quality is generally affected by the amount of concurrent subscribers on the same RF channel. LTE, being a wireless packet network, confronts challenges similar to that of a wired network where packet collisions affect performance.

It is foreseeable that densely populated areas may experience lower quality at peak usage times with a noticeable impact on performance. Channel quality is also impacted by unnecessary pings and keep-alive signals.

b. Perceived network speed

On occasion, LTE effective data rates may fall below users' expectation. By adhering to mobile development best practices, application developers can help to minimize these performance problems.

The following factors will impact the actual speeds an end user will experience:

- Distance from cell towers.
- Network congestion. As a shared medium, cellular networks typically slow down as they are loaded with users connected at the same time.
- RF interference caused by weather, terrain, buildings, and walls.
- Device specs such as OS type and processors.
- Application design.

c. Limited battery life

Batteries may have improved in capacity, but have not kept pace with power demands. LTE phones consume 5% - 20% more than prior generation phones, depending on the apps used[12]. Handset radios are the single biggest source of power drain in any device apart from the touch screen. Unlike the display, the radio is always on and is draining the battery. Apps also drain devices by not managing network connections. And with LTE devices, the radio is doing a lot more than in previous generations of devices. For example:

- **Multiple rabbit ears:** All LTE devices use MIMO (Multiple Inputs, Multiple Outputs), which essentially means a handset no longer uses a single antenna to send or receive signals. Rather, it uses two parallel transmissions. Each antenna requires its own power amplifier for transmission. So, at a basic level, two parallel transmissions is (almost) analogous to running two phones off a single battery. To compound the challenge, future devices will use even more than two antennas.
- **Scanning further:** LTE deployment is still underway, and even within an LTE market, cell towers may not be densely located. With cells spaced much further apart, devices have to boost transmission power to reach further. Since there will also be coverage gaps, smartphones will be dropping in and out of LTE coverage taxing an already over-taxed battery. So apps, especially streaming apps, need to be robust enough to run on 3G and 4G HSPA+ networks and conserve battery.

- **Packed airwaves:** Faster networks essentially mean that more bits are packed into a radio wave. But as waveform complexity increases, so too does the power needed to modulate and demodulate the radio wave.

d. Handover

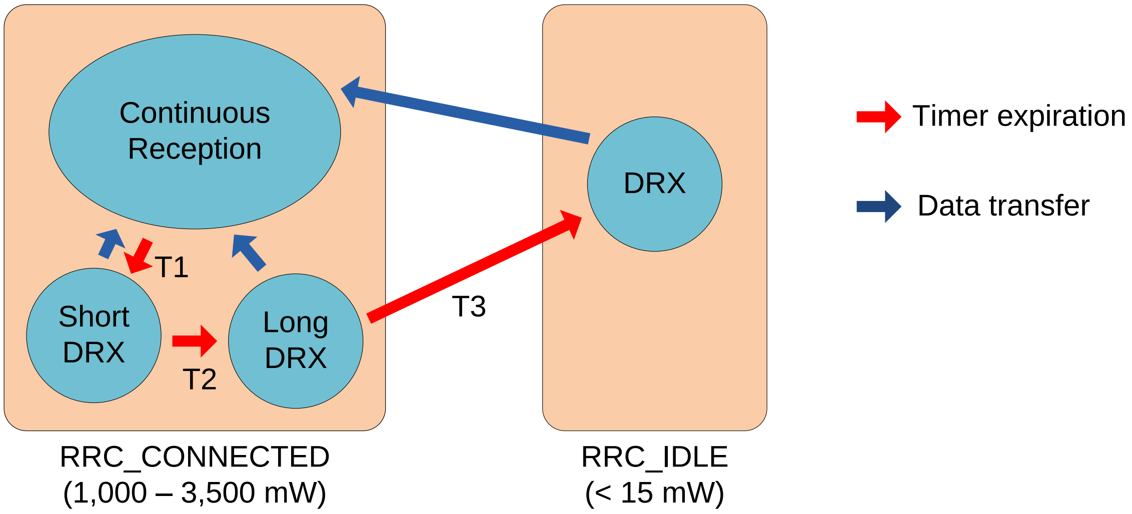
As new technologies are introduced into a carrier's network, developers need to build apps that are robust enough to manage the handover delays between a mix of 2G, 3G, 4G HSPA+, and 4G LTE environments. For instance, the handover time in a switch from an LTE environment to an HSPA+ environment is 2.4 seconds[13].

e. Signaling traffic

Signaling storms caused by applications have brought down LTE networks around the world. Signaling is the communication overhead that enables user authentication, authorization, charging, and billing[14]. Signaling costs occur when phones change their state, send data, and move from one area of coverage to another.

The addition of LTE networks increases the amount of signaling traffic. Unlike 3G's three state energy model, LTE has two states - RRC_CONNECTED and RRC_IDLE - as shown in the following diagram. Details of these LTE states are provided in the reference section of this white paper[15].

But at a high level, understanding the RRC (Radio Resource Controller) state machine for LTE will help developers better understand how to maximize the battery life. Assume for example, an LTE phone has a battery capacity of 5 Wh. An app that is in the RRC_CONNECTED state consumes 1,000 – 3,500 mW but when it is in the RRC_IDLE state it consumes less than 15 mW.



Even before LTE was added to the environment of connected devices and 3G networks, mobile operators faced significant costs in handling signaling, as opposed to the costs of handling data traffic alone[16].

Applications that disrupt a phone's radio-state management through poor networking optimization greatly increase the amount of signaling, irrespective of data size. In fact, by ignoring the implications of polling and chattiness, apps have been known to cause signaling storms and even localized network outages.

f. Third-party components

A common aspect of modern mobile app development is the use of drop-in components like advertising SDKs, analytics SDKS, and map controls. Banner ad controls have been observed to have a significant effect on uncontrolled network traffic; analytics engines and maps may be communicating with their respective back-end services outside of the direct control of the developer. Therefore, it can be useful to analyze and understand the types of traffic and the patterns of data use that these controls introduce into your app. A tool like the ARO[17] can help assess the impact of these and controls.

g. Device reachability

Users and developers both expect that devices are, in essence, permanently connected to the internet, regardless of the physical mechanisms by which this connection occurs. Platform vendors are addressing the problem of device reachability through the provision of push notifications: Apple, Google, Microsoft and RIM all provide some variation of this free service. However, push notifications have server-side limitations for developers wanting to ensure their apps can be kept synchronized and always reachable. One example of this limitation is that the push notifications are strictly one-way: there is no return path for the phone to acknowledge receipt of the message.

h. CSFB (Circuit Switched Fallback) speed impact

LTE, being an all-IP network, also has the ability to transmit voice over LTE (VoLTE) rather than using the 3G circuit switched network. AT&T has not yet deployed VoLTE, so the default method for voice is currently CSFB. However, VoLTE is the long-term goal for delivering voice, and it brings some challenges like the following:

- Enhanced 911 (E911) U.S. regulations mandate that the physical location for users dialing 911 be known to the 911 service. While that's possible for circuit-switched networks, a physical location can't be identified in an all IP network.
- LTE coverage is not available in all markets and IMS (IP Multimedia Services) network elements, a prerequisite for VoLTE, are not fully deployed.

- There are longer handover times with VoLTE. The legacy CSD has handover times of a few milliseconds, while VoLTE takes 2-3 seconds.

For these reasons and others, an LTE handset falls back to 3G circuit-switched voice and SMS services in two situations: When it is out of LTE coverage, and when a non-VoLTE voice call is made or received while in LTE coverage. This means that users will see a drop in data speeds from LTE to HSPA+ during these two scenarios. The same situation occurs when users are playing a game on LTE and a call is received.

i. IPv6 support

The capacity of IPv4, the Internet's main addressing protocol with over four billion addresses, was exhausted in most parts of the world by 2011. Because of this, IPv6, with 3.4×10^{38} addresses, replaced IPv4. IPv6 is available on non-LTE devices, but it is worth noting that while the goal is for LTE devices to support IPv6, not all handset manufacturers support NAT64 (Network Address Translation from IPv6 to IPv4). This means that developers should continue to ensure their apps are IPv6 and IPv4 compliant.

7. Best practices

This section discusses the best practices for addressing the top issues[18] concerning radio resources.

Data cap

Battery life

Speed

Robustness

a) Data cap

Due to the limited spectrum, data caps are still present in LTE data plans and users are charged for overages. Note that data caps count data uploads *AND* downloads. However, using tools such as AT&T ARO, developers can identify and work to eliminate duplicate content that can comprise as much as 20% of HTTP traffic.

As a comparison, a YouTube SD (standard definition) video on a tablet size screen (10 in) will consume 60 MB per hour while HD (high definition) video like that on YouTube HD or Netflix will consume 150 MB per hour on the same screen. So, it's clear that users can quickly exceed the cap of the most popular data plans (which are typically a few GB per month) with 50 hours of SD video or 20 hours of HD video, and thus incur overage charges.

Data consumption will obviously vary by screen size and by the types of content that users consume. To estimate daily and monthly usage for an end user, developers can use AT&T's data calculator[19].

i) Avoid large thumbnails

A 3G best practice that still applies with 4G is minimizing thumbnail sizes. A thumbnail is a miniature version of an image or a static image of a video. And the thumbnail file size is a function of four attributes:

Image file size = f (width, height, bit depth, compression)

Apps sometimes download high-resolution images for thumbnails, unaware that many devices can't display these images at that quality. This causes users to unnecessarily drain their data buckets and lose time waiting for a high-resolution thumbnail image to download, while a lower-resolution image would have saved both time and impact to their data cap.

For that reason, developers should programmatically check the device resolution and download the appropriate thumbnail for that resolution.

For example, assume a user has a download speed of 10 Mbps and downloads two, uncompressed images. Assume also, the bit depth for both images is 24 bits/pixel which is common for JPG photos.

- **Image 1**
  - Size = 3,456 pixels x 4,608 pixels x 24 bits/pixel = **382 Mb** (45.6 MB)
  - Download time = 382 Mb / 10 Mbps = **36 seconds**
- **Image 2**
  - Size = 269 pixels x 202 pixels x 24 bits/pixel = **1.3 Mb** (0.16 MB)
  - Download time = 1.3 Mb / 10 Mbps = **0.1 second**

Key lesson: Send high-resolution thumbnails to high resolution devices and low-resolution thumbnails to low resolution devices.

ii) Provide configurable data transfer settings

Always provide configuration settings for high volume data transfer scenarios such as defaulting to downloading or uploading large amounts of data over Wi-Fi. A partial list of configurable options is:

- Force Wi-Fi for specific apps or features.

- Switch to Wi-Fi rather than 3G to conserve battery.
- Alert on long downloads: It's critical to allow users the option to either configure data transfer options (Wi-Fi vs. LTE) or to be prompted before a large amount of data is transferred in either direction.
- Alert when near data limits.
- Limit or cap data quality.
- Limit or cap audio quality.
- Limit or cap overall upload and download rates.
- Limit data cache size.
- Set limits on audio or video streaming time. For example, a sleep timer for audio streaming

iii) Use Wi-Fi as much as possible[20]

Real-world LTE latencies are significantly lower than HSPA+ and in fact rival Wi-Fi speeds. From a usability perspective, users tend to feel as though a web page or video loads instantaneously when the latency is 50 ms or less.

Round-trip latency is the time spent by a user's device waiting to get a response after sending a request. So when latency matters, such as when playing lag-sensitive online games, the app can stay on LTE rather than switching to Wi-Fi, since LTE latency is around 1 - 10 ms faster than even 802.11n Wi-Fi.

However, when this slight increase in latency will not impact user experience, the app should switch to Wi-Fi (if available) since Wi-Fi avoids reducing a user's already limited data bucket. As always, tools such as ARO will help the user analyze whether their app is meeting their latency requirements.

b) Battery Life

Dual and quad-core devices with multiple antennas (MIMO) on LTE will consume power faster than legacy bands such as HSPA+ and HSPA. In fact, the first releases of LTE-ready mobile devices in 2012 and 2013 will consume even more battery power for reasons such as limited LTE coverage while LTE networks are still being rolled out.

Due to the higher amount of power used in the RRC_CONNECTED state to achieve the faster transfer speeds in LTE, battery life should represent the biggest concern for developers, and should lead them to adopt a policy of using the minimum amount of battery power needed[21].

i) Avoid constant polling[22]

Reducing polling conserves battery power. Note that frequently moving between LTE, HSPA+ and 3G networks could cause apps relying on *keepalives* or frequent updates to timeout during network handovers. Key lesson: Poll as infrequently as possible rather than leaving a channel open all the time.

Use the push notification mechanisms provided by Apple (APNs), Google (C2DM), Blackberry (Push Service) and Microsoft (MPNS) to ensure your app can refresh when necessary without relying on polling, but use push notifications with caution since depending on your app, push can actually use more battery power than polling.

For example, imagine that your email server pushes 25 emails to you in an hour. That translates into 25 push connections. However, polling every ten minutes translates into only six connections an hour. So in this scenario, polling uses less battery power than push.

Translating this into battery savings, assume that an AT&T ARO analysis shows that your app consumes 10-13 Joules (J) for each poll or email pushed to the phone.

- Poll: 13 J x 6 connections/hr = 78 J/h
- Push: 13 J x 25 connections/hr = 325 J/h
- A Samsung Captivate has a 5 Wh battery capacity
- 3,600 J/Wh x 5 Wh = 19,800 J in the battery
- Polling usage = 78 J per hr / 19,800 J x 100 = **0.4%** of the battery/hr
- Push usage = 325 J per hr / 19,800 J x 100 = **1.6%** of the battery/hr

ii) Download in bursts rather than continuously streaming[23]

Continuous streaming drains batteries faster than downloads do. In areas with slower networks, batteries are depleted faster because the device boosts its network reception in order to overcome noise.

When an app is providing a streaming service, it should warn users of the power and data costs. Key lesson: Avoid battery drain by streaming or downloading in short bursts. Using the ARO tool will graphically show how data downloads in bursts save energy.

iii) Minimize aggressive behavior

If packet loss occurs on a crowded wireless network, a device needs to back off or wait for a response. If your app is continuously attempting to regain lost packets, it can exacerbate the original failure conditions, degrading or denying service to other devices and congesting traffic. This overly aggressive behavior also drains device batteries faster. Key Lesson: An app that keeps trying to pull down signals from a congested network rapidly sucks the life from batteries.

c) Speed

There are three metrics that quantify what users mean when they say an app "feels" fast: bandwidth, latency and app responsiveness. This section focuses on principles that developers can follow to further enhance the user's feeling of using a fast application.

i) Decouple user transitions from data interactions

A big component of "feeling fast" is also communication to the user on actions: an app can feel sluggish if a tap or swipe by the user requires a data connection and response before the next UX (user experience) display. This is because users waited for a data response, even a fast data response, before the UX responded.

Decoupling, whenever possible, the UX transitions from the data interactions, will allow for a smoother UX flow and "faster" app perception to the user.

ii) Cache more often[24]

Mobile applications that allow their main operations to work even when no network is available are the most valuable. As in any distributed environment, mobile applications can benefit in various ways by maintaining a local copy of data that is unlikely to change in the short-term. By doing this, mobile applications achieve several important objectives:

- They significantly reduce repeated retrieval of duplicate data.
- They improve the app's responsiveness by accessing data locally.
- They enable limited functionality even when the app is off-line.
- They reduce battery drain by reducing overall network activity.

Reference data is a prime candidate for caching, but any kind of data that occurs within a regular time frame is also a good bet (e.g., news articles).

Caching extensively also helps eliminate spikes in network requests that can potentially cause user perceivable performance problems. Spikes in network activity typically occur at predictable times. If you understand how and when your app needs data, you'll be able to cache in advance of periods of high network activity.

iii) Flush the cache regularly

When implementing caching, also implement a sensible flushing schedule to prevent rarely-used files from consuming too much storage space.

iv) Create one session with multiple gets

Previous network generations such as 3G, due to network restrictions, were designed to handle small requests frequently. With LTE's higher bandwidth and the capability to handle longer sessions, the preference is to group requests into one or as few sessions as possible. Key lesson: Maximize bandwidth in fewer calls and group multiple requests concurrently.

v) Pre-fetch[25]

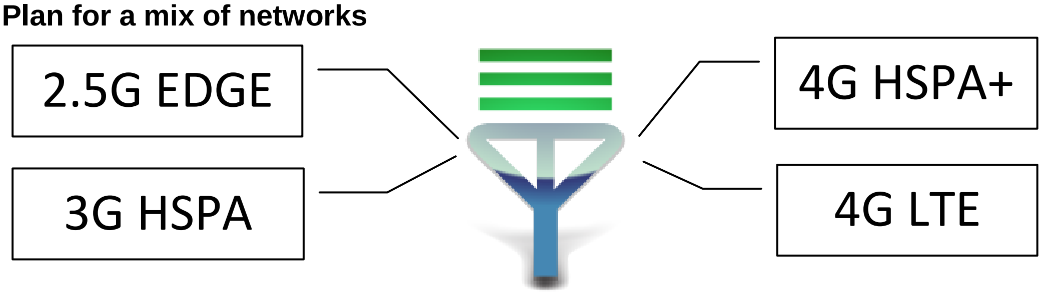
Leverage LTE's larger bandwidth and lower latency by anticipating and pre-loading data that the user wants. This requires experimenting with algorithms that work best for your app. Data retrieval during hours of expected high traffic can put your user and the network under unnecessary pressure. Consider when your application or its user may require some particular data and schedule its retrieval ahead of time.

This technique, although imperfect, provides for a better user experience. An example of where pre-fetching becomes a problem is when the app pre-fetches so much data that it is never used. This overuse drains customer's data caps rapidly in addition to using excessive memory.

d) Robustness

Apps will have to deal with multiple real-world scenarios such as users crossing dead zones, handovers between networks, and varying network speeds. Some techniques for developers to manage these scenarios is to periodically check for network type and to use bigger buffers.

i) Plan for a mix of networks



Networks differ in bandwidth, latency, throughput, reliability, and security. Even on LTE, data transfer rates will not remain constant since many factors such as network load, distance from cell towers, and terrain, impact transfer rates. This variability creates a challenge: Is the user on a fast 4G HSPA+ network or a slow LTE network? Apps will still have to make intelligent choices based on network availability (2G, 3G, 4G HSPA+, 4G LTE).

ii) Use bigger buffers

LTE is significantly faster than 4G HSPA+ and 3G, so it makes sense to use bigger buffers. Large buffers work better even when switching between networks. Buffering, unlike caching, refers to pre-loading data into a reserved area of memory. In streaming audio or video, buffering refers to downloading or streaming a certain amount of data **before** the user has consumed the data e.g., downloading some video before playing a video clip. Caching however, refers to maintaining the data, audio clip, emails, or movie, **after** the user has consumed the data.

iii) Support IPv6

As explained in the challenges section not all handset manufacturers support NAT64. So developers should continue to ensure their apps are IPv6 and IPv4 compliant. One way to do this is to ensure that apps do not use a fixed string length for IP addresses[26].

e) Follow the latest mobile developments

The mobile development community has a number of useful resources online where questions and intelligence are pooled to derive best practices. Although the generic principles provided in this white paper will mostly apply to any mobile platform, in most cases you can find specific solutions related to individual OS's and LTE standards on a number of sites like the following.

- 3rd Generation Partnership Project (3GPP) which has become the focal point for mobile systems beyond 3G (www.3gpp.org).
- AT&T Labs research (www.research.att.com).
- AT&T's Developer Program (http://developer.att.com).

f) Leverage existing tools and code samples

A number of recent initiatives have been directed at helping developers write more network-friendly apps.

- The Application Resource Optimizer (ARO) from AT&T is a diagnostic tool for analyzing mobile web application performance. AT&T ARO automatically profiles your application based on suggested best practices, and provides recommendations that allow you to optimize performance, make battery usage more efficient, and reduce network impact. Download it here.

  - http://github.com/attdevsupport/ARO
  - The github.com site contains open source repositories with multiple projects related to mobile development, including ready-to-use libraries for common programming tasks.
  - This particular branch contains the repository for Open Source ARO which lets you integrate the ARO tool and its best practice concepts into your testing and development processes.

- The Smarter Applications initiative, from the GSMA. As well as offering a prize for smart apps, these guidelines contain specific code samples based around best practices.

8. Topics to explore

There are several more best practices to explore in addition to the ones described in this white paper:

- **Legal:** The legal and ethical issues around collecting, managing and transferring end users' personal information has not changed with LTE. Developers are expected to have experts in privacy law and policy to guide them in the management of end users' information. In the U.S., the FTC (Federal Trade Commission) has brought several enforcement actions against app developers accused of misusing user data.
- **Biz Dev:** This white paper also does not cover business models like in-app advertising, in-app purchases, or freemium models. Nor does it cover which category of apps – games, utilities, social, news, etc. – are the most popular to develop. If users implement advertising in their apps, however, this paper provides the background and tools needed to optimize apps since much of the signaling in apps comes from advertising.
- **Hardware platform:** Mobile devices now not only include laptops and smartphones but tablets, e-readers, and potentially other consumer devices. This white paper is device agnostic. While this paper references primarily LTE smartphones and to a lesser extent tablets, readers should assume that statements about these devices could be extended to other consumer devices.
- **Software platform:** Developers are also assumed to be familiar with the best practices for building apps for their mobile OS such as iOS, Android, Windows Phone, and BlackBerry OS.

- **Non AT&T carriers:** This paper takes a US-centric, AT&T LTE approach. However, best practices should apply equally well to LTE or WiMax app development for other carriers like Verizon, T-Mobile, Sprint, and Clearwire.

9. Conclusion

LTE, the fourth generation mobile network, is currently being deployed nationwide. This 4G network has a significantly higher bandwidth, lower latency, better security, and more types of QoS than earlier generations. All these benefits offer the opportunity for developers to build enhanced applications not possible with 3G. Some possible applications are live, HD broadcasts from smartphones or the ability to play latency-sensitive, interactive games.

However, this network is only one of three elements in an ecosystem that has the potential for new revenue opportunities for carriers and developers. Networks, devices, and apps have to work seamlessly together to drive consumer adoption and thus revenue.

That's why this white paper focuses on technical best practices for developers to follow when building LTE optimized apps. For the same reason, non-technical topics such as business development or licensing are not addressed. These technical best practices are identical to 3G best practices for areas that customers can perceive: data caps, battery life, speed, and robustness. Developers only need to learn a few, slight modifications around LTE buffering and radio states that differ from 3G best practices. Tools such as AT&T ARO and websites such as 3GPP continue to be vital to ensure apps incorporate the best practices for app development.

Due to the vast amount of detail that could be written around each best practice, this paper intentionally stays at a high-level. However, developers are encouraged to take a deeper dive into these topics using the references section.

10. FAQ (Frequently Asked Questions) about AT&T 4G App Development

1. Are all 4G flavors identical?

No. There are several different flavors marketed as 4G such as:

4G Type	Definition	Download (Mbps)	Upload (Mbps)
HSPA+	High Speed Packet Access	3.7	1.2
WiMax	Worldwide Interoperability for Microwave Access	3.9	0.8
LTE	Long Term Evolution	23 – 61	15 - 23

Regardless of the underlying technology and marketing terms used, the key differentiators between 3G and 4G are speed and energy consumption. According to ITU (International Telecommunication Union) standards, a 4G network requires peak data speeds of 100 Mb/sec.

2. How do I test the speed of my handset's 4G network?

For a manual test, download a speed test app from your phone's app store (e.g. www. speedtest.net). Make sure you turn-off Wi-Fi before running your speed test since speed test apps measure Wi-Fi speeds if enabled.

3. How do I tell which cellular network (2G, 3G, 4G HSPA+ or 4G LTE) my handset is connected to?

Your cellular network is usually, but not always, identified by the icon next to the bars or signal indicators on your smart phone's screen.

For example, in the U.S. on AT&T's HSPA+ network with an iPhone, you may get 4G displayed. But outside the U.S., on a similar HSPA+ network, 3G may be displayed instead. Some U.S. only examples are:

AT&T Network	Icon Displayed
1G	1G or none
2G	2G or none
2.5G	G or GPRS
2.75G	E or EDGE
3G	3G
4G HSPA+	4G or H+
4G LTE	4G LTE

11. Terms and Acronyms

Acronym	Definition
1G, 2G, 3G, 4G	1ST, 2nd, 3rd and 4th generations of mobile networks
3GPP	3rd Generation Partnership Project
API	Application Programming Interface
ARO	Application Resource Optimizer
bps	Bits per second
CSD	Circuit Switched Data
CSFB	Circuit Switched Fallback
EDGE	Enhanced Data for GSM Evolution
E911	Enhanced 911
FTC	Federal Trade Commission
GB	Gigabytes
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GSMA	GSM Association
HD	High Definition
HSDPA	High Speed Download Packet Access (3G)
HSPA	High Speed Packet Access (3G)
HSPA+	High Speed Packet Access Plus is the faster version of HSPA
IMS	IP Multimedia Services
IP	Internet Protocol
IPv4	Internet Protocol with four octets (xxx.xxx.xxx.xxx)
IPv6	Internet Protocol with eight chunks (xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx)
ITU	International Telecommunication Union
LTE	Long Term Evolution
LTE-A	LTE Advanced
M2M	Machine to Machine
MIMO	Multiple Input, Multiple Output
MMO	Massively Multiplayer Online games
ms	millisecond
OS	Operating System
PCRF	Policy & Charging Rules Function
QoS	Quality of Service
RF	Radio Frequency
SDK	Software Development Kit
SMS	Short Message Service
UX	User Experience
VoLTE	Voice over LTE
WAP	Wireless Access Protocol

References

[1] Mobile apps one of three biggest drivers. http://goldengekko.blogspot.com/2010/11/mobile-apps-one-of-3-biggest-drivers.html

[2] Data Deluge. http://gigaom.com/broadband/despite-critics-cisco-stands-by-its-data-deluge/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+jkOnTheRun+%28GigaOM%3A+Mobile%29

[3] Global Mobile Data Traffic Growth. http://ibmtelconewsletter.files.wordpress.com/2011/02/wp_closing_mobile_data_revenue_gap_a4.pdf

[4] AT&T LTE speeds & latencies: http://www.lightreading.com/document.asp?doc_id=212403

[5] LTE speeds: https://developer.att.com/developer/forward.jsp?passedItemId=2400016

[6] QoS Services: http://developer.att.com/home/community/conference/LTE_Implications_of_Services.pdf

[7] PCRF examples: http://whitelassiblog.wordpress.com/2011/02/16/policy-control-and-qos-awareness-for-ims-and-lte-an-evolving-trend/

[8] U.S. rapidly becoming a nation of gamers: http://www.digitaltrends.com/gaming/study-us-rapidly-becoming-a-nation-of-gamers/

[9] AT&T's Application Resource Optimization Optimizer (ARO) is downloadable here

[10] M2M needs LTE: http://m2m.tmcnet.com/topics/m2mevolution/articles/251313-why-m2m-needs-lte-vice-versa.htm

[11] M2M needs LTE: http://m2m.tmcnet.com/topics/m2mevolution/articles/251313-why-m2m-needs-lte-vice-versa.htm

[12] Why LTE sucks (your battery): http://gigaom.com/mobile/why-lte-sucks-your-battery-that-is/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed:+OmMalik+%28GigaOM:+Tech%29

[13] Handovers from LTE to HSPA+: http://www.accuver.com/storage/downloads/AccuverResource_attdrivetestresultsandreportpreview_1326383911.pdf

[14] More Fuel for the Signaling Storm: http://lteconference.wordpress.com/2012/04/20/new-ipad-more-fuel-for-the-signalling-storm/

[15] Want to give your apps a boost? http://developer.att.com/home/community/conference/Boost_App.pdf

[16] Avoid Network Meltdowns. http://www.ospmag.com/issue/article/Avoid-Network-Meltdown

[17] AT&T's Application Resource Optimization Optimizer (ARO) is downloadable here

[18] Top wireless issues and their solutions. http://developer.att.com/home/develop/referencesandtutorials/networkapibestpractices/Top_Radio_Resource_Issues_in_Mobile_Application_Development.pdf

[19] AT&T data calculator: http://www.att.com/standalone/data-calculator/index.html

[20] Use Wi-Fi as much as possible. http://developer.att.com/developer/forward.jsp?passedItemId=9700190

[21] A Call for More Energy Efficient Apps: Minimize battery drain & reduce latencies http://www.research.att.com/articles/featured_stories/2011_03/201102_Energy_efficient?fbid=bFz29dRdhkG

[22] Reduce inefficient connections with periodic transfers. http://developer.att.com/developer/forward.jsp?passedItemId=9700174

[23] Handle multiple simultaneous TCP connections. http://developer.att.com/developer/forward.jsp?passedItemId=9700172

[24] Handle duplicate content and caching. http://developer.att.com/developer/forward.jsp?passedItemId=9700172

[25] Utilize pre-fetching. http://developer.att.com/developer/forward.jsp?passedItemId=9700172

[26] Transition to IPv6: https://developer.att.com/developer/forward.jsp?passedItemId=3100042

Note: Some developer.att.com URLs require developer account credentials in order to login and access the content.