MDSA, MULTI DECISION SCHEDULING ALGORITHM FOR UE ENERGY POWER SAVING ON MOBILE NETWORKS

Luan Ruci
FTI, Polytechnic University of Tirana
Luan Karcanaj
Kristal University, Electronic Department

Abstract

Nowadays general different kind of recurrent network applications on mobile phones are like: news feed, podcast and e-mail which mostly run in the background and are a significant source of power consumption on battery limited mobile phones. Theory of scheduling such applications by prioritizing and evaluating step by step based on many conditions (or even parameters, timers) like radio network parameters, timeout values, coverage conditions or RSSI, switching between 2G and 3G on packet data, etc is our main focus on this paper. Cellular network providers typically try to control these timeout values, though some mobile devices use a technique called fast dormancy in order to reduce the time out duration which mostly results in huge power consumption for end user. The duration of this timeout, which ranges from a few seconds to ten seconds or more, is chosen to balance the cost of signaling for resource allocation to move a radio into active state (and the resulting latency and energy costs on the device) and the wasted resources due to maintaining a radio unnecessarily in active state. We also illustrate the significant energy savings that can be achieved via scheduling of recurrent mobile phone applications considering some network parameters, conditions and also user activity time (clock time) and phone battery condition as an add. These kinds of applications such as email syncing, facebook or photo uploads can defer communication, up to a point, without sacrificing service and user perception. Other applications such as on-demand streaming can prefetch content in anticipation of future need and this will not be considered for now on our study. The other very common service for mobile users is voice and by transmitting data when a call is active brings an extension on mobile phones battery life.

Keywords: Mobile phones, RRC protocol, Energy saving, Algorithm, RSSI

1. Introduction

Since the mobile phone transmit and receive (mobile applications) based on a limited power source and provides more demanding services today, tentative to minimize the energy consumption is one of most important factors to consider in this area. In this context, the focus will be set for many areas: like modulation formats in the physical layer or the RRC-protocol in the network layer, state promotion and demotion, RX/TX levels (RSSI), parallel data connection or transfers, transferring during a voice call, considering transferring files as zip, pre-fetching and considering not to transfer during night hours or switch to 2G with EDGE mode data transfer. We will consider what have been done till now in the recent studies and analyze them for our study. As well we recognize all previous authors work and use their measurement's analysis for our study, for proposing our idea (algorithm) based on multiple checking and decision before executing any data transfer or packet calls. So for our research main goals are as: 1) evaluate the energy consumption associated with recurrent applications scheduling and shows by highlighting that there can be significant energy savings by maximizing the sleep time via batch scheduling of recurrent applications. 2) Develop a general scheduling algorithm (multi decisions) for recurrent applications and voice. All mobile phones applications and services are using the wireless air interfaces of the phone for communicating. In Figure 1 power consumption for kind of different mobile equipment services is shown. It is clear that all the capabilities (voice, data or other UE activities) using a wireless air interfaces are huge power-consumptions, so directly reducing the battery life of the phone (Perruci et al, 2009). For the moment the only way to create more powerful batteries is to make them bigger as size. However this does not well match with the evolution of the mobile terminals (nowadays Smartphone's) which plans to have less room available for the battery in order to accommodate more additional electronic components and technologies like GPS, bigger screens, other new interfaces. Understanding the nature of data transfers and behavior of much kind of applications (thousands and hundred) it's better to recall again the investigations done and use them in our proposal.

2. Motivation

Qian et al (2012) performs large-scale investigation of a particular type of application traffic pattern called periodic transfers where a handset periodically exchanges some data with a remote server every ***t*** seconds. It's seen or found that periodic transfers are generated for various reasons such as keep-alive, polling, and user behavior measurement. Periodic transfers mostly can be extremely resource-inefficient as they are small in size and short in duration relative to the periodicity. Second, periodic transfers seem to be typically delay-tolerant in that each periodic transfer instance is not initiated by a user. The typical way of generating them is to use a software timer with a fixed periodicity. Therefore, there exists some leeway for Smartphone applications to more intelligently reshape their traffic patterns to better match the characteristics of the network's radio resource management and thereby reduce their resource impact.

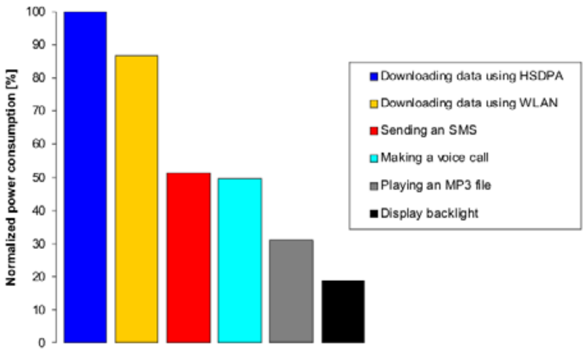


Figure 1. Power consumption for different phones services using HSDPA (Perruci, 2009)

Examples of such reshaping include adjusting the timing of each periodic transfer instance to overlap with user-triggered data transfers, which is studied also from many authors on their paper works. For finding the origins for periodic transfers by study the authors have found that they are caused by multiple factors such as *polling*, *keep-alive messages* for push-based services, *advertisement* transfers, and *user-behavior measurement*. They concluded that the vast majority of periodic transfers are small as 97% of periodic transfers are less than 10KB. Further, they are short relative to the periodicity in that 90% of them are shorter than 7 second. Understanding the origins of periodic transfers is a prerequisite for determining how to optimize such transfers without violating the application semantics. *Keep-alive Messages*, Periodic transfers are used to prevent a TCP connection from being closed by the cellular NAT, whose timeout is much shorter than the TCP timeout. *Measurements,* Mobile applications measure user behavior or preferences, and periodically send out collected information. *Polling,* Periodic polling is observed in some applications. For example, Textfree, a popular SMS application, sends an HTTP request to poll.pinger.com every 20 seconds for querying for new short messages, and usually gets a "no-new-update" response, and also there exist a couple of similar applications. Due to its high energy overhead and delayed response, such a polling-based design is clearly worse than the push-based scheme used by, for example, Facebook. *Advertisements,* Most, if not all, popular mobile advertisement platforms periodically refresh ads embedded in Smartphone applications.

Periodic DNS lookups may happen before periodic transfers described above when persistent TCP connections are not used. Although handsets have local DNS caches, some content providers may set DNS TTLs to be small for load balancing (Qian et al, 2012). So all of these activities prove to be very energy consuming in different ways and demands higher requirements on UE phones can operate in such high bandwidths without draining all energy of the mobile power source. The physical layer affects power consumption in an indirect way and with huge impact. It would be promising to utilize the UE interface in a manner that is as efficient as possible. For example by use of proper modulation and coding, a lower cost in terms of power per transmitted bit can be achieved (fast data transmission in higher modulation). Also as an important fact is that the network layer is able to reconfigure the user, by putting it on different power states, for example when no data transmissions occur the network layer may put the user in a state that consumes less energy. This is all regulated by the Radio Resource Control-protocol (RRC protocol) and will be considered on our approach. In this context, there are two important factors which determine the energy consumption due to network activity. The first one is the transmission energy, which is proportional with the length of data and transmits power level of the signal (physical layer). The second factor is the UMTS Radio Resources Control (RRC) protocol. Due to limited radio resources this protocol has been implemented in each UE to efficiently use it. The main task of RRC protocol is to introduce a state machine for controlling the user activity status as explained below.

2. RRC and Inactivity timers T1 and T2

On the RRC protocol there are four standards states used: Cell_DCH (cell dedicated channel), Cell_FACH (cell forward access channel), URA_PCH / Cell_PCH (URA/cell paging channel), IDLE. Usually PCH and Idle on many studies for similarities and easy study are considered as the same state as they consume very little power. States transition scheme as illustrated in Figure 2. The way this RRC-protocol works is simple, so in general there is need for two inactivity timers T1 and T2. If the UE receives high volumes of data (e.g. streaming video), the time between two successive packet will be very short (<T1) but if the UE receives (*DL activity*) low volumes of data the time between two successive packet will increase, if the time between two successive packets becomes higher than T1, the RRC will put the UE in the Cell_FACH state. It's the same process for the transition between Cell_FACH and IDLE state with the inactivity timer T2 (Tail time usually named). The transitions between IDLE and DCH state and the transition between IDLE and FACH state are triggered by user activity, and the state that the UE will be transitioned into depends on the Buffer

Occupancy (i.e. Data load) level in the Radio Link Control (RLC) layer. When the buffer is full, the RRC move to the DCH state. UL activity will be discussed on Data transfer size section. For Jacobsson (2012) the challenge is to configure the inactivity timers T1 and T2 (usually know as *Tail Time*). A Short time length will cause frequent reconnections that are an issue that needs to be avoided. On the other hand, prolonging the length of the inactivity timers will keep the UE in a state it does not need and will decrease the efficiency of radio resources and consumes more energy.

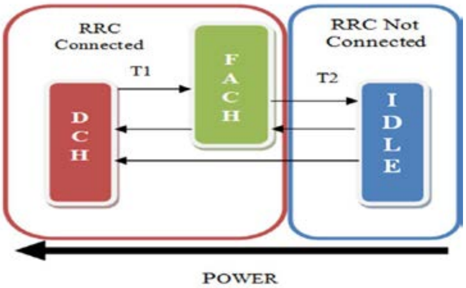
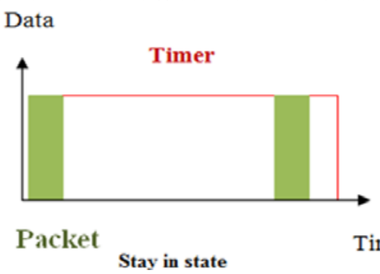


Figure 2. The radio resource control state machine (Jacobsson, 2012)

Due to the complexity of this condition, the author had chosen to trigger this transition if and only if a data request is received before the end of the last download (as per Figure 3), which will be kept under consideration on our algorithm. In the first graph of Figure 3 we can see shorter T1 is the better. A shorter T1 choice of T1 timer is a compromise between energy consumption and delay. The second graph on Figure 4 shows that a shorter T2 is more power consuming (all these were achieved by simulating and measurements done by authors). Indeed the FACH state is the more appropriate state for a web browsing utilization, if we short T2, the DCH state will be more used than the FACH state and so the power consumption will increase. Important is that by simulation Jacobsson (2012) has proved that energy consumption drops as T2 increases and then stabilizes for T2 > 10s. After RRC-protocol can be even more efficient during web browsing utilization. The simulation results confirmed that the energy consumption is at lowest at T1 = 5 seconds and T2 = 12 seconds.



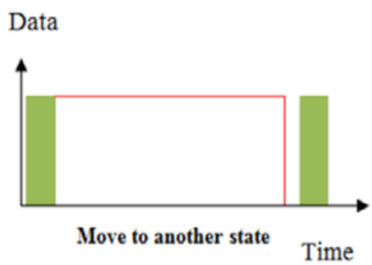


Figure 3.Proposed inactivity timer not triggered Figure 4.Proposed inactivity timer triggered

For Liu et al (2011) if tries to aggregate small transmissions into large ones so that the occurrence of Tails (and thus energy consumption) can be reduced. The aggregation is achieved through prefetching (prefetch the data likely to be requested in the future for news, video, etc) and delayed transfer or batching (defer the delay-tolerant data to be transferred later for Email, RSS, Flickr, Dropbox, etc). If there are delayed past transmissions or future requested transmissions, it would be beneficial to schedule them. This mechanism will be called again in subsection of synchronization concept. The prefetching and delayed transfers are beneficial to the transmissions for a number of applications, including email, RSS, news, software updates and video. Intuitively, it is much better to perform several bursty transfers at once with a single ramp and tail cost than to have several ramp-transfer-tail cycles (Qian et al, 2010). When these recurrent applications are left to schedule themselves independently of other recurrent applications, they can have a very negative impact on battery life by bringing the phone out of sleep mode an unnecessary number of times (Brittan). The key idea is to maximize sleep time from a holistic view of repeating intervals and resource usage of all active recurrent applications and thereby achieve energy savings. Simply Batch scheduling algorithm works based on discovering and by setting the biggest existing interval (of one of recurrent applications) for next transfers.

3. Data transfer sizes impact

In the RRC-protocol, the state transition from FACH to DCH state is triggered when the Buffer Occupancy is full (like the standard RRC protocol) and if the user wants to download a data of size larger then for example M. When the download is completed the RRC moves back to the FACH state. On CELL FACH: In this state the cellular interface shares a communication channel to the NodeB. It can access the random/forward access (CELL RACH/CELL FACH) common channels. Its data rate is only a few hundred bytes per second. CELL RACH is an uplink channel and CELL FACH is a downlink channel (Nurminen, 2010). As mentioned previously another aspect of cellular radios is the tail energy overhead which is closely related to data transfers sizes and switches from one state to another.

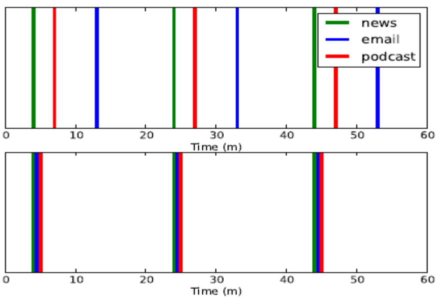


Figure 5. Applications batch scheduling (Brittan)

Transition from CELL FACH to CELL DCH is triggered by changes in the downlink/uplink queue sizes maintained for these two states in the radio network controller or RNC. Schulman et al (2010) measurements indicate state transition thresholds of 151 bytes for the uplink queue and 119 bytes for the downlink queue (note that this is Network operator's dependence). Once either queue size exceeds its threshold, CELL DCH is entered. All this approaches will be included latter on our proposed algorithm combined by priority put on decision when they can take place (for transmitting). Algorithm for discovery state transitions and fetching T1 and T2 would be a good idea to be used on our algorithm on a later stage or version of this algorithm. For example trying to ping from UE with different kind of packet size and monitoring on the same time the UE and measuring the timeout and packet sizes which make the differences.

4. Signal level and power control mechanism

Another factor which influences the power, the time, and therefore the energy, is the received signal strength. As shown in Figure 6 the power levels increase as well when the received signal strength decreases and so, more energy is needed when the signal is weaker (Perruci et al, 2012). On UMTS / HSPA (3.5 G) the most used access techniques is CDMA. CDMA is interference limited multiple access system. Here all users transmit and receive on the same frequency (in case of FDD mode), so internal interference are generated by the system and are the most significant factor in determining system capacity and call quality.

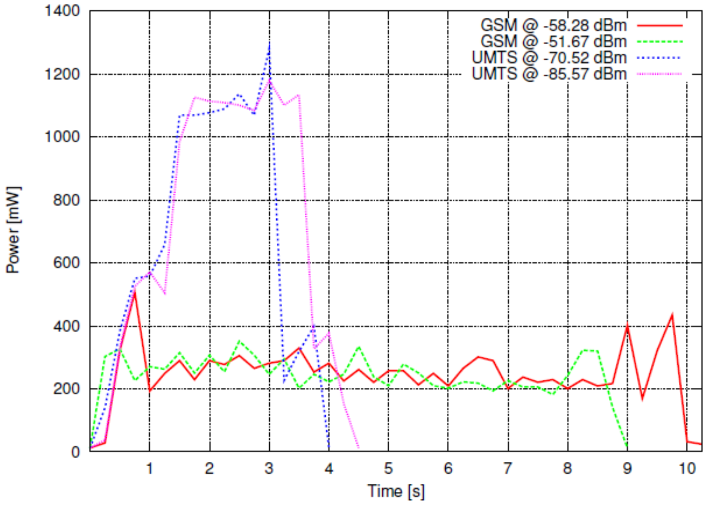


Figure 6. Power traces for sending 200 bytes using UMTS & GSM with different RSSI (Perruci, 2009)

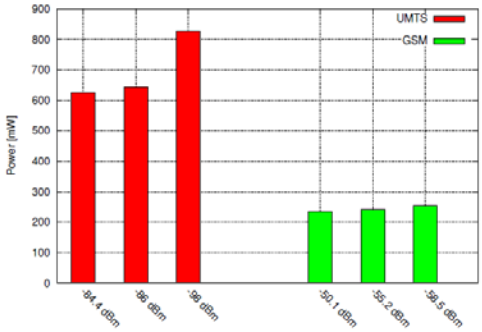


Figure 7. Mean power level for sending 200 bytes using UMTS.

On UMTS / HSPA (3.5G) the most used access techniques is CDMA. CDMA is interference limited multiple access system. Here all users transmit and receive on the same frequency (in case of FDD mode), so internal interference are generated by the system and are the most significant factor in determining system capacity and call quality. As the MS moves around, the RF environment continuously changes due to fast and slow fading, external interference, shadowing, and other factors. This is a complicated process and many authors on their studies consider the UE in static state (not on move during their tests) for simplicity of their tests. The aim of the dynamic power control is to limit transmitted power on both the links while maintaining link quality under all conditions. Additional advantages are longer mobile battery life and longer life span of BTS power amplifiers. In reality UE is measuring the RSSI. The problem when measuring the electromagnetic fields emanating from mobile phone base stations is that the output power level of the traffic channels varies according to traffic load. This means that the field strength also varies. However, at least one channel per base station in GSM and UMTS networks outputs at a constant, known power level. This is the BCCH (broadcast control channel) in GSM and the P-CPICH (primary common pilot channel) in UMTS. The signal strength has a direct impact on radio energy consumption, which is a significant component of overall energy consumption on mobile devices. Cellular radios consume more power and suffer reduced data rate when the signal is weak. According to measurements done, the communication energy per bit can be as much as *6x* higher when the signal is weak than when it is strong. To realize energy savings, applications must preferentially communicate when the signal is strong, either by deferring non-urgent communication or by advancing anticipated communication to coincide with periods of strong signal. Schulman et al (2010) had developed an energy-aware scheduling algorithm for different workloads synchronizing and streaming and evaluate these via simulation driven by traces obtained during actual drives, demonstrating energy savings of up to 60%. First, the power, i.e., energy per unit time, drawn by the radio increases when the signal is weak. This is because a weak signal often means that the mobile device is near the edge of coverage and hence it must amplify the received signal. Also, it must transmit both upstream data and acknowledges for downstream data, at a higher power for power control reasons or simply to have its feedback be heard by the tower. Second, the data rate, i.e., bits per unit time, decreases when the signal is weak. The reason is that the radio typically switches to a lower rate of modulation to keep the bit error rate low. For the author these signal variations can be exploited by preferentially communicating when the signal is strong, thereby saving energy. This is taken on consideration on our proposed algorithm as one of main contributors. For authors the signal strength must be predicted so that opportunities for energy-efficient communication can be anticipated, while taking application deadlines into account. Here we do share another approach, as we consider by measuring the RSSI for a period of time like (3 sec average) and decide based on signal strength when to transmit or not. Communicating when the signal is strong reduces the energy cost by cutting both the power drawn by the radio and the communication time. Communication in strong signal takes less power, for both transmission and reception, although. A strong signal also makes feasible advanced modulation schemes that yield higher throughput. The radio draws more current to operate in low signal locations. One reason is that the power amplifier switches to a high power mode to counter the drop in signal strength. This applies not only for transmission, but also for reception since the mobile clients continuously reports the received signal strength to the base station, 800 to 1600 times per second (the base station uses this feedback to choose an appropriate modulation and data rate). Communication at a poor signal location can result in a device power draw that is 50% higher than at good signal locations. Moderate signal strength values (-90 to -70 RSSI) are more common than extremes. So as a conclusion when the signal is weak, not only does data transfer take longer to complete, but the radio is also simultaneously consuming higher power. These two factors are cumulative, so the overall energy required to transfer a fixed chunk of data, i.e. energy per bit, can be as much as six times higher (25% throughput and 50% more power) while communicating from poor signal locations compared to good signal locations. On the real field highest and lowest strength values are -50 and -120 RSSI and there are frequent signal strength variations between -90 and -70 RSSI. The cost of communicating at -90 instead of at -70 RSSI entails the use of about 20% additional power (Figure 5) and a median throughput that is 50% lower. This result in energy per bit on the Pre that is 2.4 times higher at -90 RSSI compared to at -70 RSSI. Thus, if applications were to preferentially communicate at -70 RSSI instead of at -90 RSSI, the potential communication energy savings are 60% (Nurminen, 2010).

4.1 Downlink v.s Uplink

On everyday activity users receives and send data, so simultaneously they use both UL and DL. For radio planners the Uplink determines coverage (especially for low traffic loads) and Downlink determines maximum throughput of the cell. In general one of the main ideas is to maximize throughput and capacity as the mobile equipment power drain will be reduced as explained before. Data throughput is generally limited by the downlink and may be shared between all users. By many test for Nurminen (2010) it is also visible that *uploading* consumes more energy than *downloading* (18% more with WLAN and 10% more in 3G). Also to our knowledge there is no any study related differences between 2 different modes of operations on UMTS: TDD and FDD data transfer point of view. Usually for many authors the most interesting had been the DL on testing since the data send on UL is small for example Http requests and so on, so the main focus had been there.

4.2 HOM, High order of Modulation

To achieve transmission of digital data by air the data bits needs to be processed into symbols and then be carried by a sinusoidal wave carrier, this is done by using a so called modulation. A basic explanation of modulation is to map a certain amount of bits into symbols to increase the transmission rate and gain spectral efficiency. There are different modulation formats used depending on service and country. The most common formats used for mobile telephony are the Quadrature Amplitude Modulation (QAM) and Phase Shift Keying (PSK) formats. HSDPA releases use was released with the modulation 16QAM. This modulation carries 4 bits per symbol and enables an increased downlink speed up to 7.2 Mbit/s. The latest release of HSPDA that use 64QAM-modulation, which progress carries 8-bit per symbol and supports a theoretical rate up to 84 Mbit/s. In reality its radio conditions “define” the modulation to be used, so here not so much to determine for our algorithm. This has a very close relation to RSSI.

4.3 Synchronizations concept with RSSI consideration

As previously mentioned nowadays there are hundreds and thousands of applications in use for mobile users. Not all of them are friendly applications for Network operators by signaling point of view, causing issues for operators and UE itself by reducing the battery. The first application class that Schulman et al (2010) consider is *background synchronization*, where the device probes a server periodically to check for new mail, updated news, or similar pending messages. Within each periodic interval, the syncing operation can be scheduled whenever the signal is strong like 5 bars, and they put set of a threshold of approximately -75 RSSI, such that syncs performed when the signal is above the threshold consistently consume less energy than when below the threshold. The average sync at a location with signal above -75 requires 75.3% of the energy when the signal is below -75 as per Figure.8 (Nurminen, 2010). Above -75 RSSI, the sync energy is consistently low. Below this threshold, the sync energy is higher and variable. By choosing to perform sync more often when the signal is good, actual data communication (e.g., downloading of emails and attachments), if any, would also likely happen during periods of good signal, resulting in additional energy savings.

4.4 Streaming concept with RSSI consideration

Streaming sites typically transmit pre-generated content over HTTP. In addition, some of these sites throttle the rate at which the data is streamed to the client, while keeping the client-side buffer non-empty to avoid playout disruptions. Video playing is the worst case as the most energy consuming power hardware on the devices. In general, energy consumed at -93 RSSI is about 50% higher than the energy consumed at -73 RSSI.

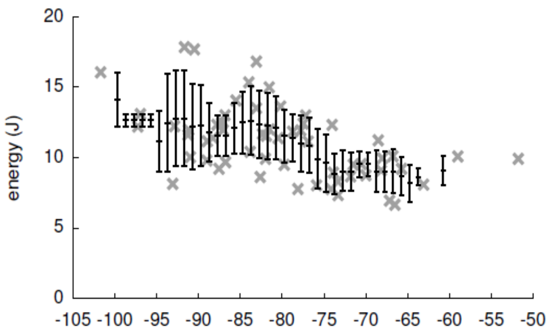


Figure 8. Signal Strength and Energy consumption (Nurminen, 2010)

However, in some tests of Nurminen (2010), energy consumed at -73 RSSI is higher than energy consumed at -93 RSSI, because of lower bandwidth at -73 RSSI for this experiment (perhaps due to competing/shared users). Authors here are proposing a scheduling streaming algorithm. They look how to efficiently schedule slots. A slot is defined as the period of time where a single frame can be transmitted. Since data rates are not fixed, each slot can be of variable width depending on the expected datarate at that time. The power consumed to transmit a frame in a slot is also variable. Given N frames and M slots, where N $\leq$ M, the optimal scheduling problem seeks to find an assignment for each frame to one of the slots, such that the total energy required to transmit N frames is the minimum of all possible assignments (Nurminen, 2010). One main idea is to greedily schedule the frames in the best N slots which incur the least energy for communication. However, this approach ignores the cost of tail energy incurred every time there is a communication. When multiple frames are scheduled in consecutive slots, the entire batch of frames incur only one tail, as opposed to a tail for each frame if they are spaced out in time data stream of size S over certain duration of time T with minimal energy. To make the problem tractable, they divided the input stream into fixed size chunks of N frames, and time T divided into.

5. Shared medium and effect on data througput

Many recent studies have quantified the service-level stability of ubiquitous wireless networks. They found that bandwidth changes over time, even though signal strength measurements remain stable. Usually even though signal strength measurements were stable at a location, network bandwidth measurements vary by almost 50%. On a stables signal strength at a location reason that could be affecting the bandwidth are: many users under the same NodeB / Wcell’s are having data activity (download or uploading) causing a high traffic by reducing capacity offered for rest of users, applied QoS streaming for different users (RNC feature) have a direct impact as specific UE’s have different profiles for data, any bottleneck on any interfaces of Network operator (overload), or any remote server issues. Some of these aspects will be considered on our proposed Algorithm. By preventing or deferring transmitting when the throughput is slow (on the most of cases during peak hours and when the serving area is full of users) it can save the battery and reduce its consumption. There can be some options how this can be identified and one of them is to ping and check the TTL. This ping can be started on itself as a background application during only predefined time (Peak hours) testing, gathering and analyzing them for the use at multi-decision algorithm.

6. Parallel connection (data and voice)

With the increasing number of mobile services, opportunities to transfer data in parallel will increase. Parallel transfer can happen by accident but it can also be an engineering goal if it results into energy savings. In both 3G and WLAN cases the energy per bit decreases significantly when the bitrate increases (as one shoot transfer). This for many authors suggests a strategy for energy saving. The higher the bitrate we are able to use in the communication, the better for the battery duration for UE. A possibility to increase the bitrates and thus transfer data in more energy efficient fashion is to execute multiple communication actions at the same time. In this way we do not need to influence the speeds of the individual connections but we are still be able to take advantage of the more energy efficient data transfer enabled by higher bitrates. Many options as below:

6.1 TCP connections during a Voice call

In case of a cellular voice call, these components are constantly drawing power because the system has no time to enter a low power sleep mode, as the voice stream requires steady transfer of data. However, the voice stream requires only a small part of the available bandwidth. A TCP connection running in parallel with the cellular voice call can use the spare bandwidth but as all the communication circuitry is already powered on to serve the voice call, it needs very little additional power. The slight increase in power consumption can be attributed to the increased use of CPU and memory. The thick gray/green curves on Figure 9 shows a case when both voice call and data download are active simultaneously. On the experiments made by Nurminen (2010) the power consumption is around 1.7W and 1.4W when the display light is on and off respectively. From the combined curve, when data download is performed during a voice call the power consumption is only slightly higher than what the voice call would anyhow require. In this measurement, the extra power is only 10%. The data transfer time increases slightly up to 12% during a parallel voice call. As a result, if a data transfer during a voice call takes place, the user is able to perform the same activity with only 23% of the energy that it would take without the voice call. Figure 9 shows power consumptions when a) a voice call is active, b) a data download is active, and c) both voice call and data download are active at the same time (Nurminen, 2010). VoIP measurements are also interesting because there are two connections carrying IP traffic: UDP based VoIP and TCP based data transfer. Therefore, it is possible that the similar gains can be achieved in other applications based on UDP such as in media streaming.

6.2 Parallel TCP Connection's

The results showed to Nurminen (2010) that for the parallel TCP case the energy savings depend on the situation in a complicated way. On 3G parallel connections created over 20% savings. Connection for downloading and at the same time another TCP connection for uploading to see if the two-directional traffic would change the situation. Interestingly, in this case the results were different.

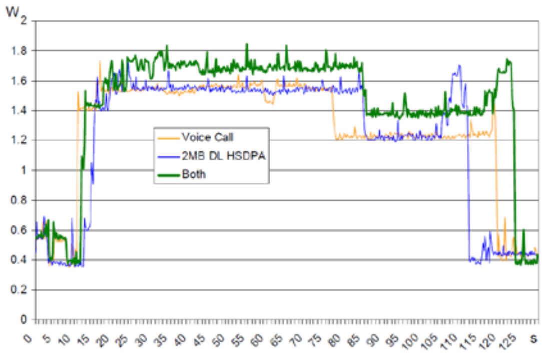


Figure 9. Power consumption for different conditions (Nurminen, 2010)

In 3G it's seen by tests the opposite effect; parallel download and upload required 24% (this for FDD systems) more energy than the two parallel downloads. So this approach is skipped on our algorithm. Finally, the case when we multiple interfaces for TCP download are used, the energy consumption does not support this approach resulting into 40% to 50% higher energy consumption than the use of a single interface (so using different interfaces like WLAN, Bluetooth etc is not considered for now on our algorithm). The use of multiple interfaces does not seem to be useful due to impact on increasing power consumption form other side. So parallel connections save energy but the gains vary depending on the technology. TCP downloads during 3G voice calls result into 75% - 90% energy savings, TCP downloads during VoIP calls result into 30% - 40% savings, and TCP downloads when other TCP streams are active at the same interface result into 0% - 20% savings. But as tested, if two streams experience the same internet or server bottleneck there is no gain. Second, if there are two slow to medium speed streams that do not fulfill the radio channel completely then sending them in parallel reduces the energy consumption.

7. 2G and 3G for Voice calls and Idle status

Many authors of present results of power and energy consumption measurements conducted on mobile phones for 2G and 3G networks. The services under investigation were text messaging, voice and data. The paper reports larger energy consumption in 3G networks for text messaging and voice services than energy consumption in 2G networks (our interest is for Voice and Data). On the other side the 3G networks become more energy friendly when large volumes of data have to be downloaded (this can be concluded also from the summary we have made till now from analysis done by other authors mentioned). The results imply that mobile phones should switch the network in dependency of the service used to save the maximum amount of energy. As this handover consumes energy, we include its analysis in our idea. As per phones info the battery duration of mobile phones highly depends whether they are connected to GSM or 3G networks, the results show that making a call using GSM costs 46% less energy and receiving a call costs 50% less energy than using UMTS. Being idle while connected to a GSM network costs 41% less than UMTS. But from a user perspective, it makes more sense to be all the time connected to the GSM network and switch to 3G only if data connection is needed. In the energy consumed in that specific scenario by using 2G alone, 3G alone and the intelligent switching between the networks. A significant reduction in terms of energy when using the intelligent switching can be seen. To simplify this method for energy conservation, it is possible to develop a middle-ware solution that handles the switching depending on the used applications in an automated fashion. Sometimes this is difficult as many mobile operators prevent switching from 3G to 2G by parameters at least for Data part.

8. Battery info for Power-save mode

There are many features or tools to enter the UE on the power-save mode when some conditions are meet. The idea of the power save mode is to easily allow the user to save battery capacity when the battery charge level is low. The three things that the power-save mode does in order to reduce the power consumption of the device are: 1) dimming the screen 2) closing the internet connection and 3) forcing the phone to GSM network. These power-saving actions are activated by running some Symbian system command's (for Nokia Mobiles). When the power-save mode is exited, the display brightness is set back to the brightness level before entering the power-save mode and the radio access technology is set back to ”dual mode”. Internet connections are not re-established. On this approach here is to include a timer to check every period for example on hourly the battery status and after 40% level reached, status to be checked every 5 minutes till battery reach 25 % for entering on PSM or forced - idle. Also there exist options to switch off other interfaces like WLAN, Bluetooth etc.

9. UE screen status

Usually the user on surfing or doing other PS activates on their mobiles do not work all the time, for example there is inactivity time during night or for example when they are busy with other thing and mobile is left not used or at least downloading or having other recurrent applications running on background. In such cases would be beneficial from using the screen mode change process and time (clock) to decide if entering in PSM, switching to 2G or 3G mode. So you get 3G when the screen's on, and then after a while this utility automatically drops you back to 2G, for background email etc without needing the higher speed and power drain.

10. Compression Tools

Many authors have shown the benefit of using compression to save energy when sending short messages on mobile phones MMS or even sending Emails with attachments. The energy spent for compressing and decompressing messages is much less than the gain obtained by sending less data. In order for this to happen efficiently, the data first needs to be compressed using a data compression application. Data compression algorithms like Lempel-Ziv 77 (LZ77) and Lempel-Ziv-Welch (LZW). With lossless data compression algorithms compression and decompression can be efficiently implemented on a mobile device, even with the hardware limitations such as low processing power, static memory and battery life. Lossless data compression has many advantages on a mobile device, such as reducing the network bandwidth required for data exchange, reducing the disk space required for storage and minimizing the main memory required (Brittan). Using such compression algorithms for emails with attachments or photo and video uploads with big sizes can show many benefits. This is going to be considered on latter stages of our algorithm.

11. MDSA Algorithm proposed

So in our proposed algorithm would be good idea to avoid transferring during peak hours (to be considered during optimization phase) and zipping attachments when transferring via emails or other application using attachment. Also good level of RSSI is beneficial for users on move as it changes and it can decide when it wants to transmit (for sure also static users benefit from it). Scheduling by 2 to 3 scales when signal drops will make possible to switch between 2G and 3G or even in PSM mode. Signal measurements come at no extra energy cost because the phone's cellular protocol needs them for handoff. Recall that for sync, our goal is to put the processor to sleep for a calculated interval, so that the phone wakes up when it is at a strong signal location. As well by considering activity time we put some “restrictions” which are coordinated with other checks in parallel.

Module: 0

0. Packet data status

M0.a) 3G with HSPA & 2G with EDGE [dual mode – no restrictions].
M0.b) 2G only with EDGE/GPRS.
M0.c) 2G with no EDGE/GPRS.
M0.d) PSM (M0.c & screen off or reduced & Interfaces disabled).

Module: 1

1. Decide based on Battery status (L1 &L2)

M1.a) Check battery status every 1 hour till condition M1.a. 2 are meet and report as L otherwise goto M1.a. 1.

M1.a. 1) If battery level (L1) ≤ 35% goto step M1.b & M0.b otherwise goto M1.a. 2.

M1.a. 2) If battery level (L2) ≥ 35% & M5.a. 2 meet goto M0.a & M4.d otherwise M1.a. 1.

M1.b) Check battery status every 5 minutes till condition M1.c are met otherwise goto M0.b.

M1.c) If battery level (L3) ≤ 15% goto M0.c or M0.d and generate alarm and note for charging till M1.a. 1is meet otherwise go M1.a.

Module: 2

2. Decide based on the Clock time

M2.a) If time t1 between 00:00 to 07:00 goto M1.a. 1 otherwise M2.a.1.

M2.a.1) If time t1 between 00:00 to 07:00 and M1.a. 2 conditions are meet goto M0.b otherwise go to M2.a.

M2.b) If time between 07:00 to 00:00 (not T_1) & M5.a. 2 goto M1.a. 1 if ok go to M4.b. 1 otherwise M2.b.1.

M2.b. 1) If time between 07:00 to 00:00 (not T_1) & M5.a.1meet goto M1.a. 2 and if meet goto M0.b otherwise M2.b.

M2.c) change t1 based on preference.

Module: 3

3. Decide based on the UE screen activity

M3.a) If the screen is off goto M2.a otherwise M2.b.

M3.b) If the screen is on goto M1.a.1and check for M4.a otherwise goto M4.b

Module: 4

4. Decide based on the user activity (CS or PS call)

M4.a) Check if active voice call (MO or MT call) and if yes goto M1.a.1 and M4.b otherwise M0.b

M4.a. 1) Check if M4.b meets and if not got to 4.b.1

M4.a. 2)1 if no active voice (CS) calls (MO or MT call) call for M4.b

M4.b) Check if active PS calls (UL or DL PS data) and if yes got M5.a.1

M4.b. 1) force transferring delayed TCP/UDP connections for recurrent applications and goto

M4.c & M4.d if new add applications done from.

M4.c) Set T1 =5s and T2 =12 s if M4.b are meet otherwise got to

M4.d) Discover applications periodic time and Set periodic transfer to one of 2 options:

M4.d. 1) check every 1 hour if M4.b meets.

M4.d. 2) Application max periodic time seen if new applications are added.

Module: 5

5. Decide based on the user RSSI values

M5.a) Measure RSSI (R) values and set it to R every 1 sec and on 5th sec do average of Ra.

M5.a. 1) If Ra value ≤ -90 RSSI stop planned/queued transfers or M0.c and deferring transferring till M5.a.2 are meet otherwise M5.a.2.

M5.a. 2) If -90 ≤ Ra ≤ -70 is meet goto M1.a.1 and M4.b.1

(For Ra ≤-70 also cases Ra ≥ -75 are covered)

Module: 6

M6.a.) Press button to deactivate conditions or consider always M0.a [No restrictions]

Conclusion:

This is a first step toward a power save algorithm. Future work will consist on involving some real test and experiment by testing on mobile platforms Like Nokia E6 (e.g., Symbian phones). This will show the performance and need for improvement and changes in the proposed algorithm. We know that imposing a restriction on specific intervals is unsuitable for many important types of recurrent applications and can be frustrating from user perspective point of view but in the next work this will be the main focus to optimize. Also in the future works will try to optimize the algorithm to conclude transfers based on their size so considering channels state on 3G like DCH and FACH.

References:

Feng Qian, Zhaoguang Wang, 2012. Periodic Transfers in Mobile Applications-Network-wide Origin, Impact, and Optimization. *WWW 2012 Session: Mobile Web Performance*.

Gian Paolo Perrucci, Frank H.P. Fitzek and Giovanni Sasso, 2009. On the Impact of 2G and 3G Network Usage for Mobile Phone's BatteryLife. *European Wireless 2009*.

Hao Liu, Yaoxue Zhang and Yuezhi Zhou, 2011. TailTheft Leveraging the Wasted Time for Saving Energy in Cellular Communications. *MobiArch’11*.

Niclas Jacobsson, 2012. Energy Consumption in Wireless Networks Progress Report.

Aaron Schulman, 2010. Bartendr a Practical Approach to Energy-aware Cellular Data Scheduling. *MobiCom'10.*
Feng Qian and Zhaoguang Wang, 2010. Characterizing Radio Resource Allocation for 3G Networks.
Jukka K. Nurminen, 2010. Parallel Connections and Their Effect on the Battery Consumption of a Mobile phone. *IEEE CCNC 2010.*
Paul Brittan. Evaluating Lossless Data Compression Algorithms for use on Mobile Devices. *Literature Synthesis.*

Appendix: MDSA conditions general overview.