# The case for context-aware resources management in mobile operating systems

Narseo Vallina-Rodriguez and Jon Crowcroft

**Abstract** Efficient management of mobile resources from an energy perspective in modern smart-phones is paramount nowadays. Today's mobile phones are equipped with a wide range of sensing, computational, storage and communication resources. The diverse range of sensors such as microphones, cameras, accelerometers, gyroscopes, GPS, digital compass and proximity sensors allow mobile apps to be context-aware whereas the ability to have connectivity almost everywhere has bootstrapped the birth of rich and interactive mobile applications and the integration of cloud services. However, the intense use of those resources can easily be translated into power-hungry applications. The way users interact with their mobile handsets and the availability of mobile resources is context dependent. Consequently, understanding how users interact with their applications and integrating context-aware resources management techniques in the core features of a mobile operating system can provide benefits such as energy savings and usability. This chapter describes how context drives the way users interact with their handsets and how it determines the availability and state of hardware resources in order to explain different context-aware resources management systems and the different attempts to incorporate this feature in mobile operating systems.

## 1 Introduction

Lithium-ion battery technologies have not experienced the same evolution as the rest of hardware components in mobile handsets. The battery capacity is limited by

Narseo Vallina-Rodriguez
Computer Lab, University of Cambridge, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD. UK e-mail: Narseo.Vallina-Rodriguez@cl.cam.ac.uk

Jon Crowcroft
Computer Lab, University of Cambridge, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD. UK e-mail: Jon.Crowcroft@cl.cam.ac.uk

design factors such as size and weight, thus the only alternative left at the moment to extend their battery life is reducing the power consumption at the hardware level and designing more energy efficient applications and operating systems. However, despite the recent achievements in improving the energy efficiency by both hardware and software vendors, mobile handsets still suffer from severe energy limitations. As the consumption of energy can be attributed to the use of particular hardware components (mainly sensors, displays and wireless interfaces), there is a clear need to discover new ways of reducing the use of such components without compromising the user experience and services delivered by mobile applications.

Generally, there's a energy-usability trade-off when managing networking and sensing resources in mobile systems. Typical software energy saving techniques aim at keeping hardware resources in low power mode for as long as possible. However, transitions between power modes can imply an energy cost depending on the power features of the resource. As an example, cellular interfaces present three power modes: DCH (Dedicated Channel) FACH (Forward Access Channel, an intermediate power mode popularly known as *"tail-energy"*) and IDLE. In the case of WCDMA technologies, a large fraction of energy is wasted in these intermediate but still high-power states after the completion of a typical transfer in case there is going to be an immediate transmission once the current one is finished in order to improve the user experience in cellular networks. In GSM technologies, the time spent in the FACH state is much smaller compared to 3G (6 vs. 12 secs) [1]. In fact, these transitions are typically related to applications running on the device and the interaction patterns of the user [2] [3].

On the other hand, the quality of resources such as cellular networks can vary depending on the location, time of the day and even season of the year [4]. Users tend to run a specific set of application (and consequently, they access an specific set of hardware resources) depending on the social or personal activity they are performing. Consequently, this dependency has implications in the energy consumption of the handset since both the state of resources and the way users interact with their handsets and applications are social and context-dependent. As an example, a mobile user can experience frequent periods of network blackouts in certain locations (specially when moving) so launching a network-intense video streaming application in these situations might not be the best idea.

Incorporating contextual information and energy-awareness as a key feature of mobile operating systems has been barely explored despite its enormous potential. In this chapter, we will show how mobile operating systems can exploit contextual information to adapt the system to the environment and the users' needs in order to extend the battery life of the handset without compromising the users' experience. In other words, the operating system can learn from user's interaction and mobility patterns to know what kind of resource is likely to be demanded by the user at an specific context and the state of these resources. In section 2 we will show how users interaction is driven by context while section 3.1 describes the dependency of resources availability (e.g. wireless interfaces and location sensors) with context. Finally, section 4 describes two ongoing projects that are trying to incorporate context-aware features at the operating system level to manage resources: CondOS [5] and ErdOS [6].

## 2 Are users' interaction with their handsets driven by context?

Several studies have tried to explore the impact of contextual information on mobile systems. As an example, *LiveLab* is an event-based resources logger for jail broken iPhone devices used to measure real-world smartphone usage and wireless networks [7]. Despite the fact that the results obtained are not statistically representative, they indicate that both users' interaction with the device and the state of the resources depend on contextual factors such as time and space [8].

Vallina-Rodriguez *et al.* performed a study using a background application to collect traces directly from 18 mature Android users during 2 weeks [2]. The dataset contains contextual information and more than 25 state and usage statistics from multiple resources and applications, sampled every 10 seconds. This analysis uses machine learning techniques to understand the dependencies between resources caused by users interaction and both spatial and temporal context. The paper demonstrates that energy demand and resource availability depend enormously on each participant's pattern of usage both in terms of which applications they ran and when and where they were doing so. This interaction can be very variable and dynamic both in time and space.

Spatial context affects how users interact with their handsets. Figure 1 shows three scatterplots of the average percentage of daily usage of the 3G interface, telephony and the screen versus its standard deviation while the users are subscribed to their three most popular cells. Users *U1*, *U5*, *U8*, *U9*, *U14* and *U18* have a strong routine due to their low variance and are quite likely to interact with certain resources in those locations. These users present a more predictable interaction pattern than other users who are likely to interact with their resources in non-frequent location and in transitions between them (e.g. while commuting).

However, temporal context also provides useful hints about how resources are used. Figure 2 plots the average usage and availability of different mobile resources such as battery, telephony, network, screen and CPU for ten users per hour of day. Each one of the x-axis bins represent an hour of the day and the colour indicates their averaged value during duration of the experiment. These results reveal that battery usage, charging opportunities and power limitations are well defined for some individuals in the temporal domain while others are more random. For instance, users such as *U2* and *U5*) and yet others present a much burstier pattern for specific resources.

While most resources available in mobile handsets can be recovered and reallocated once used by a process, energy can be only recovered when the user manually charges the handset. As a consequence, an energy-aware operating system must be able to estimate when energy will be consumed, how much energy will be available and when it will be recovered by predicting future charging opportunities

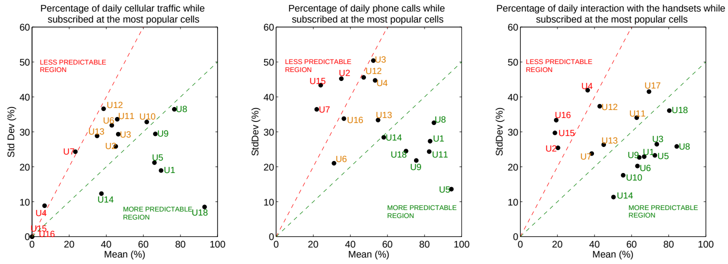


Fig. 1: User classification by their percentage of the usage/interaction with the 3G interface, telephony service and screen while subscribed at the most common cells (likely to be users' workplace and home). The *x* axis represents the daily average usage and the *y* axis the standard deviation. This information can be used to identify the places where the energy consumption will be higher and also to infer the predictability of the user interaction and the state of a resource.

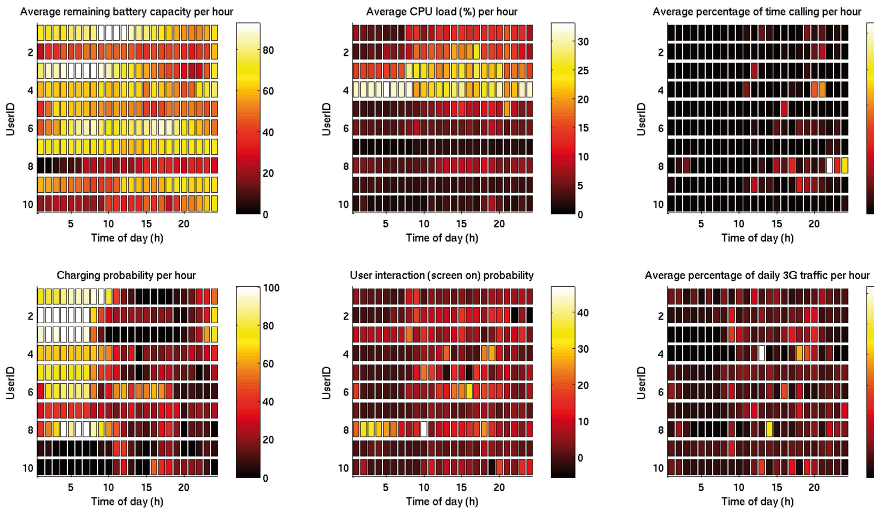


Fig. 2: Average usage and availability of different mobile resources for users *U1* to *U10* per hour of day. As in Figure 1, this information can be used to identify peaks of usage and temporal patterns on those resources.

and their uncertainty. Charging actions are in fact context-dependent and relatively predictable. Oliver [9] used classification methods to identify three distinct types of charging patterns among a large dataset of 17.300 Blackberry users. Those clusters are defined as *"opportunistic chargers"*, *"light-consumers"* and *"nigh time chargers"*. In their results, they evaluated that it is possible to predict the energy level on a mobile handset within 7% error within an hour and within 28% error within 24 hours.

Figure 3 shows the *correlograms* or *autocorrelation plots* of the battery capacity and the CPU load for three users for a 7 day lag. Note that a correlogram is a plot of the sample autocorrelations versus the time lags. This kind of analysis helps to identify randomness and periodicities in a dataset. The correlogram clearly reveals that *U3* presents a clear charging periodicity of 24 hours approximately while *U8* does not have such a marked routine. However, those results highly depend on the resource analysed. As we can observe, the CPU load is not periodic at all indicating that CPU load might be more difficult to predict than battery capacity. This confirms that an efficient resources management technique must be user-centric and must try to identify the randomness, patterns and predictability of each individual user and device.

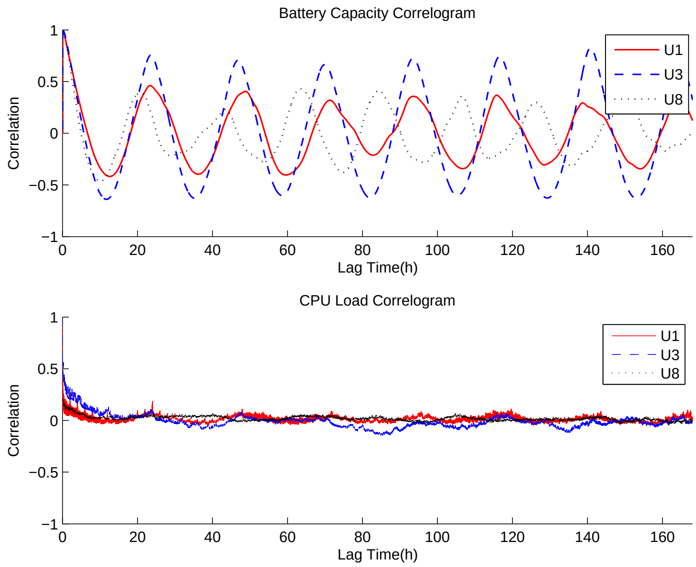


Fig. 3: Correlogram for the CPU load and the battery capacity for users *U1*, *U3* and *U8* during a period of 7 days. The battery capacity correlogram shows a clear pattern and a periodicity on the energy consumption and recharging cycle every 24 hours approximately while the CPU load is highly random.

Nevertheless, Banerjee *et al.* in [10] claim that, despite the fact that there is a great variation among users, most of recharges happen when the battery has substantial energy left and a considerable portion of the recharges are driven by context (e.g. location and time). In a similar way, Ravi *et al.* proposed a system for context-aware battery management that warns the mobile user when it detects a power limitation before the next charging opportunity is going to happen [11]. This stops the system from compromising the execution and performance of crucial applications and services such as telephony and messaging by non-crucial ones. This system uses the current set of applications running, the battery discharging rate[1] and phone call logs as inputs of their forecasting algorithms. The results indicate that their algorithm can predict battery consumption and charging opportunities only for users with a low usage entropy. The main difficulty is predicting phone calls because of their dependency of the almost unpredictable social factors and the variability of calling patterns between weekends and weekdays.

As a conclusion, contextual information can be used to understand how mobile users interact with their devices. This can enable innovative ways to manage resources as we will see in the following sections. However, the entropy of users' interaction patterns and habits require identifying new techniques to efficiently leverage this information without impacting negatively on the user experience. The operating system can infer which applications are likely to be executed by the user at a given context and which resources might be required by them. As a result, the operating system can proactively pre-load these applications and set the hardware resources that will be required by them in the right power mode while turning off (or setting in low-power modes) those hardware components and applications that are not likely to be accessed.

## 3 Context-aided mobile resource management

In addition to users' interaction patterns with mobile handsets, the state and availability of mobile resources depend on contextual aspects. Two clear examples are:

- *GPS*. The number of visible satellites by the receiver affects the time to fix their location from the *cold-start* [2] phase and its accuracy. However, satellites are constantly moving in their orbits, the number of satellites visible for the receiver depends on time of day and the day [13]. Other aspects such as reflections and radio obstacles also affect the time required by the receivers (thus the energy required) to fix their location. Nevertheless, in the case of assisted-GPS, chips can vary depending on the availability of a cellular network to quickly access the ephemeris of the satellites [14] thus reducing considerably the time to fix the location.
- *Cellular interfaces*. The energy consumption of cellular interfaces and their quality of service depend on the receiving signal strength of the radio link [15]. As the *signal-to-noise ratio* (SNR) increases, more retransmissions at the link layer

[1] Battery discharging rate might arguably not be the best indicator to measure energy consumption in mobile handsets. This signal is very noisy since it depends on hardware and users' habits and requires complex methods to be properly calibrated [12]

[2] If the GPS chip has not been used in a long time, then the *Time To First Fix* (TTFF) can be longer because it needs to download the satellites ephemeris and almanac before it can make the calculations. Usually, the GPS-receiver also needs 4 satellites to accurately fix its location. This is usually referred to as *cold start*. In cases when the chip was recently used (in the order of minutes or even few hours), the time to fix would be even faster (i.e. *warm start* and *hot start* phases).

are required and therefore, more energy is consumed. As we can see in Figure 4, co-located nodes present different network coverage and quality depending on the location and the mobile operator. The signal strength is in fact, context-dependent. As any radio technology, the quality of the link can be affected by other aspects such as provider's network deployment, whether the node is indoors or outdoors, node's mobility, radio obstacles and radio interferences [16].

### 3.1 Wireless interfaces

Mobile handsets present different wireless interfaces that range from cellular networks such as GPRS to LTE and IEEE 802.11 (i.e. WiFi) technologies. The key differentiators between these interfaces are their availability and their power states. The operating system could switch the type of network depending on which service is being requested by the user and the applications. The OS can select the optimal link for a wireless communication taking into account the energy-delay trade-off and applications requirements [1], [17], [18]. As a result, the system can adapt to channel conditions by leveraging contextual [19], local and historic information to decide whether and when it must defer a transmission in order to save energy.

As we can see in network coverage maps collected by crowd-sourcing means such as *OpenSignalMap* [20], the network availability and quality depend on location. *3GTest* [21] is a cross-platform application that checks the state of cellular networks and the performance of network-based applications. The traces from 30.000 mobile users all over the world confirm the impact of contextual aspects on the performance of cellular networks. Network properties can vary depending on the time-of-day and location for a specific operator, as Tan *et al.* [22] had also previously demonstrated for a more geographically limited environment such as Hong Kong. A detailed knowledge of the network properties can help to identify bottlenecks in wireless network and also performance limitations and bugs in hardware, operating system and popular network-centric applications. In fact, the latency of the radio link depends on the current power state of the wireless machine. Based on historic data, the operating system can seamlessly enable caching mechanisms to applications accessing wireless interfaces and also supporting traffic shaping techniques to adapt the applications' traffic to the conditions of the wireless interface at a given location.

On the other hand, the availability of WiFi access points is reduced to specific locations as we can see for London city center (UK) in Figure 5. IEEE 802.11 networks usually present a lower latency than cellular networks for transmitting data but they present a higher cost when the device is associating to the access point. As a consequence, reducing the energy cost of scanning and associating to the access point is essential. Because of this reason, most of the works described in this section try to leverage contextual information to smartly wake up the WiFi interface from sleep mode when it is likely to have an access point.

(a) Operator 1 (b) Operator 2

Fig. 4: Signal strength perceived by two identical co-located handsets in several locations in west and centre of Cambridge (UK) with different network operators. Lighter points indicate better signal strength.

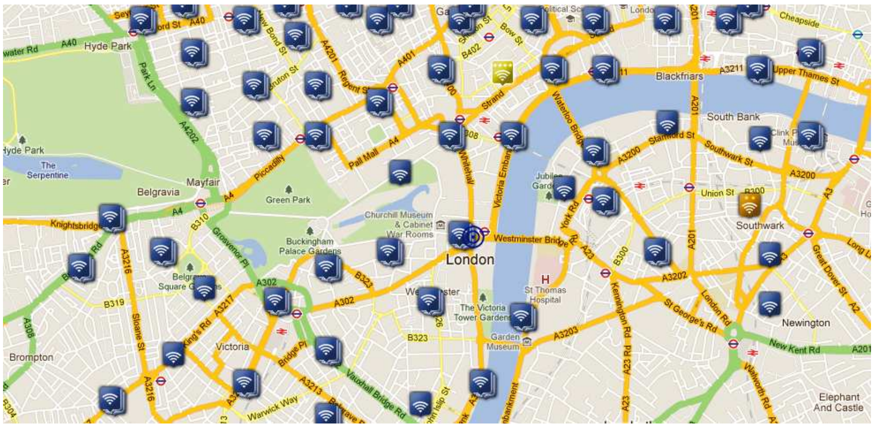


Fig. 5: Open IEEE 802.11g Access Points in London city center (UK). Snapshot obtained from WiFi Map UK [23]

The operating system can adapt the AP's discovery enquiries to minimise the energy consumption while maximising the chances of having connectivity. The works by Agarwa *et al* [24] and *Blue-Fi* [25] are two good examples to illustrate this claim. These papers describe how to save energy by exploiting other resources such as Bluetooth radios and contextual information to serve as a paging channel for IEEE 802.11 technologies. The results show that it is possible to save between 23% to 48% of energy compared to the present IEEE 802.11 standard operating modes with negligible impact on performance. The system can predict when there will be Wi-Fi connectivity by combining contextual information obtained from Bluetooth scans contact-patterns and cell-tower information. Likewise, *Context-for-wireless* is a context-aware intelligent switching algorithm between WiFi and cellular networks to reduce the energy consumption substantially [26]. *Context-for-wireless* leverages contextual information such as time, historic data, cellular network conditions and mobility to formulate the selection of wireless interfaces as a statistical decision problem and to predict future network conditions.

### *3.2 Location Sensors*

Mobile applications tend to become context-aware. By simply looking at the applications market of mobile platforms such as *Android* and *iPhone*, it is possible to find a large number of context-aware applications and location-aware online services such as *Google Maps* and *Foursquare* [27]. Applications often need location data to update locally relevant information, to provide a service requested by the user and also to find nearby friends and places of interest.

Modern smartphones include different types of location sensors with different resolution and energy demands such as cellular network-based location providers, WiFi-based, A-GPS (Assisted-GPS), gyroscopes and compass. New location techniques are being investigated such as audio fingerprints [28], signal-strenght fingerprints [29]. and geo-magnetism fingerprints [30]. Other solutions leverage phone sensors, audio beaconing infrastructures and opportunistic user-intersection (in space-time) to develop an electronic escort service formulated like routing packets in Delay-Tolerant Networks (DTNs) [31]. All these technologies are mainly focused on providing more efficient indoor localisation. However, most context-aware applications are based on standard sensors. They usually prefer A-GPS over its alternatives (e.g. network-based location providers such as *Skyhook* [32] and *Location-api* [19]) because of its accuracy despite its higher energy cost. Cellular network based location services present a mean error in the order of 300m (can be in the order of several km) and, given a location they can report different locations because of radio link changes. As a consequence, the research community tried to find solutions to save energy when accessing location information without sacrificing accuracy. [33] describes four alternative techniques to GPS sensing to reduce the energy consumption:

- **Substitution and Suppression** makes use of alternative location-sensing mechanisms (e.g. network-based location sensing or combined use of accelerometers and compass) that consumes less power than GPS. Substitution decides when to use more energy-efficient sensors instead of more energy-costly ones such as GPS. As a consequence, the system can automatically decrease the energy consumption of mobile sensing applications. On the other hand, suppression utilises less power-intensive sensors. As an example, it is possible to use accelerometers to suppress unnecessary GPS sensing if the user is static.

- **Piggybacking** synchronises the location sensing requests from multiple running location-based applications[3].
- **Adaption** adjusts sensing parameters such as time and distance depending on the remaining battery capacity. This technique tries to find heuristics to adapt the sampling rate without sacrificing accuracy.
- **Probabilistic Models**. Some methodologies rely on probabilistic models of users' location to infer future locations to reduce the number of sensing reads.

Continuous location-sensing can be very costly in terms of energy. Several research projects tried to combine in a different way those techniques, mainly looking at the energy-accuracy tradeoff as it is summarised in Table 1:

| Sensor-based optimisations | | | | | | |
|---|---|---|---|---|---|---|
| | Sensors Used | | | | | |
| *Name* | *GPS* | *Accel.* | *GSM* | *Piggybacking* | *Probabilistic Models* | *Adaptation* |
| **EnLoc** | ✓ | ✓ | ✓ | | ✓ | |
| **A-Loc** | ✓ | ✓ | ✓ | | ✓ | |
| **EnTracked** | | | | | ✓ | ✓ |
| **RAPS** | ✓ | ✓ | ✓ | | ✓ | ✓ |
| **Zhuang** | ✓ | ✓ | ✓ | ✓ | | ✓ |
| **Caps** | ✓ | | ✓ | | ✓ | |

Table 1: Location sensing optimisations. Most of the works aimt to tackle the continuous location sensing challenge by combining different techniques. This table highlights the different methodologies used by each one of the solutions and the sensors they are using.

*EnLoc* [35] provides a location sensing adaptive framework that exploits mobility patterns of the user and decides which sensor to use taking into account the accuracy-energy trade-off of the different location sensors available in mobile phones. The authors take advantage of users' *Logical Mobility Tree* (LMT). This model allows sampling at a few uncertainty points which may be sufficient for predicting future locations. EnLoc utilises dynamic programming to find the optimal localisation accuracy for a given energy budget: it decides which localisation sensor will be the best one for a given scenario and energy budget.

Similarly, *EnTracked* [36] estimates and predicts the system state and mobility of the user [4] to schedule position updates in order to minimise the power consumption while optimising robustness. *EnTracked* uses the GPS-estimated uncertainty to

[3] The energy consumption becomes even more significant if multiple applications are requesting location reads independently. [33] is the only one that applies this technique. Android OS *Location Providers* follow a similar philosophy [34]

[4] The system only supports pedestrians as possible movement model and uses accelerometer to infer users' mobility

quickly schedule a new measurement if a potential bad measurement is performed. Other solutions exploit Hidden Markov Models to predict the mobility of the users and they also take advantage of Bluetooth scans to identify static scenarios based on devices in the same location [37]. A more sophisticated version of the system was recently proposed in [38]. In this case, they use sensors such as radio fingerprints, accelerometer and compass with the collaboration of a server to estimate the time to sleep of the GPS receiver before the next positioning sensing.

*A-Loc* [39] incorporates probabilistic models of user location and sensor errors. It was implemented as a middleware solution for Android devices which requires applications' collaboration. A-Loc selects the most energy-efficient sensor to meet applications accuracy requirements which must be either specified explicitly by applications or automatically by the system. The system uses the probabilistic models to choose among different localisation methods and tunes the energy expenditure to dynamically meet the error requirements.

Other systems such as *RAPS* (Rate adaptive GPS-based positioning for smartphones) [40] take inspiration from the observation that GPS accuracy in urban areas can be poor due to moving objects, trees' shade and building reflections. To solve this issue, RAPS uses location-time history of the user to estimate user velocity and adaptively turn on GPS in case the estimated uncertainty in the prediction exceeds the accuracy threshold. RAPS presents three different approaches: it allows synchronising GPS readings between neighbouring mobile devices to reduce power consumption, it blocks GPS reads when the user is subscribed to cellular base stations where it is unlikely to get a GPS read (e.g. an area where the user is usually indoors) and it exploits accelerometer data to estimate user velocity. It also proposes sharing position readings among nearby devices using Bluetooth in order to further reduce GPS activation. However, RAPS is mainly designed for pedestrian use, and a significant portion of the energy savings come from avoiding GPS activation when it is likely to be unavailable. The authors recently proposed newer approaches such as [41]. In this case, they try to combine the accuracy and energy complementary features of GPS and network-based solution. This paper is based on the observation that users exhibit consistency in their everyday routes, having a sequence of Cell-IDs. The system can provide an accurate estimation of user's position by monitoring the cell-ID transitions and using a history of GPS readings obtained within a cell. They use the Smith-Waterman algorithm for sequence matching between similar historic data. They look for a sub-sequence in the database that matches and they pick up the sequence that matches the best and they turn ON GPS when there is no good matching. However, such system has the limitation of not being able to detect small detours in common routes.

## 4 Context-aware mobile operating systems

In previous sections we have seen that the most commonly used hardware components by applications in mobile phones are context-dependent in terms of availability, energy cost as also users' interaction patterns. Current mobile operating systems are also multitasking. By executing **ps** in the terminal of an Android handset we can identify more than 60 processes running simultaneously. Many of these processes are context-aware and they are accessing shared resources such as sensors and wireless interfaces. However, battery capacity is still the main limitation in mobile systems and, as a consequence, mobile devices are likely to experience power limitations at any time depending on how intensely the users interact with them. Two mobile operating systems already aim to leverage contextual information to prolong the battery life on mobile handsets: CondOS [5] and ErdOS [6].

CondOS has been conceived after observing that context-awareness is already a reality in modern mobile platforms and applications. Mobile handsets support a diverse range of sensing hardware and they are capable of executing the algorithms required to process raw sensed data. However, the way contextual information is generated and provided to applications can be more efficient by integrating context-aware resources management techniques in the operating systems. If applications manage and generate their own context independently, the power consumption can increase. It is necessary to provide a central content provider that coordinates all the context requests and the operating system is the right place for that. They consider that raw-sensed data must be converted into *"contextual data units"* (CDUs) by the operating system. A CDU is defined as a higher level data abstraction compared to the current contextual data provided by modern mobile platforms. Those objects contain a unit of meaningful context data to applications such as *walking* or *commuting*). The authors also list the potential benefits that can be achieved with a context-aware OS:

- *Memory Management*. Actions such as *"running"* and *"walking"* may suggest the user to load a music player or a workout app. On the other hand, actions such as *"driving"* may suggest loading a navigator. The operating system can see how users' interacted previously with applications in order to pro-load them and set the hardware resources in the right power mode to improve the user experience.
- *Scheduling*. Context information can help to schedule processes while limiting the impact on battery life and user experience. CondOS suggests that context can directly influence process priorities based on the users' preferences and the applications that are likely to be executed at a given location.
- *I/O*. Contextual data can help to adapt notifications such as the ringing mechanism or the appropriate input method to the situation (e.g. voice search features might be useful while the user is walking but they might not be the best choice in a noisy environments or in the opera). The operating system can also adapt manage wireless interfaces aided with contextual information as we have already seen in Section 3.1.
- *Security*. Security can be adapted to the location. For example, security requirements can be relaxed at home, enabling interaction and sharing data with other devices in the home network. On the other hand, in public places the security policies can be more rigid in order to reduce the potential security and privacy risks.

- *Energy savings*. As we have seen in the previous sections, mobile operating systems can predict future charging opportunities from contextual information. The operating system can manage applications and resources in order to meet the energy goals that users' interaction might impose. Moreover, having a central source of contextual information can potentially save energy by reducing the number of requests to the hardware resource. Applications can collaborate and share interests on resources in a similar fashion as Android OS does with its *"location providers"* [34].

Mobile operating systems need to make efficient and autonomous allocation decisions whilst maximising the users experience. Software should guarantee energy efficiency in addition to the traditional OS perspective of maximising performance [42]. Recently, energy-aware operating systems attracted the attention of the research community again with mobile operating systems such as Cinder [43] and ErdOS [6]. However, those two projects follow different philosophies about how energy management should be performed, and by whom.

Cinder follows the philosophy of ECOSystem [44] and Odyssey [45]. They try to leverage the interaction between applications and operating system without necessarily being context-aware. In the case of Odyssey, applications adapt to the available energy and resources to provide different quality of service to the users in runtime while ECOSystem fairly allocates energy shares to multiple hardware components and applications. *Cinder* [43] allocates energy to applications using two abstractions called *reserve* and *taps* to form a graph of resource consumption. When an application consumes a resource, the Cinder kernel reduces the right values in the corresponding reserve and its scheduler only allows threads to run if they have enough reserves to run. The rate at which the reserves are being consumed is controlled by the *taps* (a special-purpose thread whose only job is to transfer energy between reserves at proportional or constant rates). Once an application has consumed all its reserves, the kernel prevents its threads to perform more actions. Nevertheless, Cinder allows *reserve debits* between tasks for performing additional actions. Note that most of the modern mobile OSs usually give priority to foreground processes over the rest of the apps and non-system background processes in order to improve the user experience and also to prolong the battery life.

A different approach is followed by *ErdOS* [6]. This operating system[5] does not require interaction and communication means between applications and OS. It is completely seamless to applications. ErdOS also leverages contextual information to manage resources efficiently customised for each user. ErdOS was motivated by the observation that resources' state (e.g. GPS and cellular networks) and the usage patterns and habits of mobile users are diverse and highly context-dependent. As mobile systems present energy peaks caused by periods of high interaction from the users, managing and allocating computing resources to applications proactively based on predictions of the resources state and the users' demands is more flexible and efficient than algorithmic resources management. In order to support this feature, ErdOS monitors resources state, applications resource demands and users'

[5] ErdOS is conceived as an Android OS extension

interaction patterns with applications. It learns from users' behaviour and habits (defined as the *users' activity* abstraction) to predict the future resources demands and the resources availability in an event-based fashion. In fact, users generally remain subscribed to a small set of base stations and the majority of interaction with their resources or applications takes place there. ErdOS builds a location-based model of resources usage and resources demands per location in order to predict power limitations and peaks of energy consumption. Such a model might help to detect malware and buggy applications by identifying situations where resources demand are out of the norm.

Additionally, the authors consider that computation should not be exclusively limited to local resources. Accessing resources in neighbouring handsets opportunistically can be beneficial both in terms of energy and usability by enabling access to resources that have the right power mode [46]. By considering the social activity of mobile phone users, we can see that large portions of a user's daily life are spent in close proximity of other mobile phone users with devices that incorporate similar hardware resources. Indeed, if we consider a commuter travelling by bus and using a location-based service on her mobile phone, there is a high probability that a significant number of co-commuters are also using their phone's GPS and cellular networks to interact with similar services. Additionally, in social events such as music concerts or sport events, large numbers of co-located users may use their phone to access the internet simultaneously. This enables more opportunities for sharing resources opportunistically and, as a consequence, more opportunities to reduce the energy consumption. As a consequence, ErdOS tries to exploit this opportunity for improving the energy efficiency of mobile phone usage while making acceptable compromises in the QoS, by trying to aggregate, share and coordinate resources of multiple users at close proximity. Nevertheless, contextual information can play an important role in making ErdOS even more energy efficient by allowing the system to adapt the resources discovery enquiries and the privacy and security policies to the probability of discovering devices at a given location.

## 5 Summary

Mobile handsets are power-hungry devices because of the integration of power-hungry hardware resources such as touchscreen displays and location sensors. Moreover, they support Internet data services anytime (almost) anywhere so they are always connected to the network. All those resources bootstrap a rich ecosystem of mobile applications but their design is clearly driven by usability factors rather than energy efficiency. However, managing mobile resources from an energy-efficient perspective without diminishing the user experience is clearly one of the most challenging problems in mobile computing nowadays. Power management considerations often require certain actions to be deferred, avoided or slowed down to prolong battery life. In this chapter, we have seen that contextual information can be a useful source of data to manage hardware resources more efficiently in mobile systems. It can allow the operating system to dynamically predict the power states of the hardware components and applications behaviour at a given location. However, those techniques can impact on the user experience with the handsets and there is still an important work to be done in this space.

## References


1. N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: a measurement study and implications for network applications,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC ’09, (New York, NY, USA), pp. 280–293, ACM, 2009.
2. N. Vallina-Rodriguez, P. Hui, J. Crowcroft, and A. Rice, “Exhausting battery statistics: understanding the energy demands on mobile handsets,” in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld ’10, (New York, NY, USA), pp. 9–14, ACM, 2010.
3. I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, “Measuring serendipity: connecting people, locations and interests in a mobile 3G network,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC ’09, (New York, NY, USA), pp. 267–279, ACM, 2009.
4. Q. Xu, A. Gerber, Z. M. Mao, and J. Pang, “AccuLoc: practical localization of performance measurements in 3G networks,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, (New York, NY, USA), pp. 183–196, ACM, 2011.
5. D. Chu, A. Kansal, J. Liu, and F. Zhao, “Mobile Apps: It’s Time to Move Up to CondOS,” *USENIX HotOS*, 2011.
6. N. Vallina-Rodriguez and J. Crowcroft, “Erdos: Achieving Energy Savings in Mobile OS,” in *Proceedings of the 6th ACM International Workshop on Mobility in the Evolving Internet Architectures*, MobiArch’11, 2011.
7. C. Shepard, A. Rahmati, C. Tossell, L. Zhong, and P. Kortum, “LiveLab: measuring wireless networks and smartphone users in the field,” *SIGMETRICS Perform. Eval. Rev.*, vol. 38, pp. 15–20, January 2011.
8. H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, “Diversity in smartphone usage,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys ’10, (New York, NY, USA), pp. 179–194, ACM, 2010.
9. E. Oliver, “Diversity in Smartphone Energy Consumption,” in *ACM workshop on wireless of the students, by the students, for the students*, 2010.
10. N. Banerjee, A. Rahmati, M. D. Corner, S. Rollins, and L. Zhong, “Users and batteries: interactions and adaptive energy management in mobile systems,” in *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp ’07, (Berlin, Heidelberg), pp. 217–234, Springer-Verlag, 2007.
11. N. Ravi, J. Scott, L. Han, and L. Iftode, “Context-aware Battery Management for Mobile Phones,” in *PERCOM ’08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, (Washington, DC, USA), pp. 224–233, IEEE Computer Society, 2008.
12. M. Dong and L. Zhong, “Self-constructive high-rate system energy modeling for battery-powered mobile systems,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, (New York, NY, USA), pp. 335–348, ACM, 2011.
13. M. Wing, A. Eklund, and L. Kellogs, “Consumer-grade global positioning system (GPS) accuracy and reliability,” in *Journal of forestry*, vol. 103, pp. 169–173, 2005.

14. G. M. Djuknic and R. E. Richton, “Geolocation and Assisted GPS,” *Computer*, vol. 34, pp. 123–125, February 2001.
15. R. Chakravorty, S. Katti, J. Crowcroft, and I. Pratt, “Flow Aggregation for Enhanced TCP over Wide-Area Wireless,” in *in Proc. IEEE INFOCOM*, pp. 1754–1764, 2003.
16. X. Chen, H. Zhai, J. Wang, and Y. Fang, “A Survey on Improving TCP Performance over Wireless Networks,” in *In Resource Management in Wireless Networking, Cardei M, Cardei I, Du D-Z (eds*, pp. 657–695, Kluwer Academic Publishers, 2005.
17. M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, “Energy-delay tradeoffs in smartphone applications,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys ’10, (New York, NY, USA), pp. 255–270, ACM, 2010.
18. C. Pluntke, L. Eggert, and N. Kiukkonen, “Saving mobile device energy with multipath TCP,” in *Proceedings of the sixth international workshop on MobiArch*, MobiArch ’11, (New York, NY, USA), pp. 1–6, ACM, 2011.
19. Location-Api. http://location-api.com/.
20. OpenSignalMap. http://opensignalmap.com/.
21. J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl, “Anatomizing application performance differences on smartphones,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services* , MobiSys ’10, (New York, NY, USA), pp. 165–178, ACM, 2010.
22. W. L. Tan, F. Lam, and W. C. Lau, “An Empirical Study on the Capacity and Performance of 3G Networks,” *IEEE Transactions on Mobile Computing*, vol. 7, pp. 737–750, June 2008.
23. “Wifi Map UK.” http://www.wifimapuk.com/home/.
24. Y. Agarwal, C. Schurgers, and R. Gupta, “Dynamic power management using on demand paging for networked embedded systems,” in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, ASP-DAC ’05, (New York, NY, USA), pp. 755–759, ACM, 2005.
25. G. Ananthanarayanan and I. Stoica, “Blue-Fi: enhancing Wi-Fi performance using bluetooth signals,” in *Proceedings of the 7th international conference on Mobile systems, applications, and services* , MobiSys ’09, (New York, NY, USA), pp. 249–262, ACM, 2009.
26. A. Rahmati and L. Zhong, “Context-for-wireless: context-sensitive energy-efficient wireless data transfer,” in *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys ’07, (New York, NY, USA), pp. 165–178, ACM, 2007.
27. FourSquare. https://foursquare.com/.
28. S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, “Indoor localization without infrastructure using the acoustic background spectrum,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, (New York, NY, USA), pp. 155–168, ACM, 2011.
29. C.-W. You, P. Huang, H.-h. Chu, Y.-C. Chen, J.-R. Chiang, and S.-Y. Lau, “Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization,” *Ad Hoc Netw.*, vol. 6, pp. 1221–1237, November 2008.
30. J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, “Indoor location sensing using geo-magnetism,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, (New York, NY, USA), pp. 141–154, ACM, 2011.
31. I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, “Did you see Bob?: human localization using mobile phones,” in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom ’10, (New York, NY, USA), pp. 149–160, ACM, 2010.
32. SkyHook Wireless. http://www.skyhookwireless.com/.
33. Z. Zhuang, K.-H. Kim, and J. P. Singh, “Improving energy efficiency of location sensing on smartphones,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services* , MobiSys ’10, (New York, NY, USA), pp. 315–330, ACM, 2010.
34. Android Developers. http://developer.android.com/reference/android/location/LocationManager.html.

35. I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox, “EnLoc: Energy-Efficient Localization for Mobile Phones,” in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, no. 4, pp. 2716–2720, IEEE, Apr. 2009.
36. M. B. Kjaergaard, J. Langdal, T. Godsk, and T. Toftkjaer, “Entracked: energy-efficient robust position tracking for mobile devices,” in *Proceedings of the 7th international conference on Mobile systems, applications, and services* , MobiSys ’09, (New York, NY, USA), pp. 221–234, ACM, 2009.
37. T. Farrell, R. Cheng, and K. Rothermel, “Energy-efficient monitoring of mobile objects with uncertainty-aware tolerances,” in *Proceedings of the 11th International Database Engineering and Applications Symposium*, (Washington, DC, USA), pp. 129–140, IEEE Computer Society, 2007.
38. M. B. Kjaergaard, S. Bhattacharya, H. Blunck, and P. Nurmi, “Energy-efficient trajectory tracking for mobile devices,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, (New York, NY, USA), pp. 307–320, ACM, 2011.
39. K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, “Energy-accuracy trade-off for continuous mobile device location,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services* , MobiSys ’10, (New York, NY, USA), pp. 285–298, ACM, 2010.
40. J. Paek, J. Kim, and R. Govindan, “Energy-efficient rate-adaptive GPS-based positioning for smartphones,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services* , MobiSys ’10, (New York, NY, USA), pp. 299–314, ACM, 2010.
41. J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, “Energy-efficient positioning for smartphones using Cell-ID sequence matching,” in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys ’11, (New York, NY, USA), pp. 293–306, ACM, 2011.
42. C. S. Ellis and M. Watt, “Every Joule is Precious Energy in Computing,” *ACM SIGOPS*, 2000.
43. A. Roy, S. M. Rumble, R. Stutsman, P. Levis, D. Mazières, and N. Zeldovich, “Energy management in mobile devices with the cinder operating system,” in *Proceedings of the sixth conference on Computer systems*, EuroSys ’11, (New York, NY, USA), pp. 139–152, ACM, 2011.
44. C. S. Ellis, “The case for higher-level power management,” in *Proceedings of the The Seventh Workshop on Hot Topics in Operating Systems*, HOTOS ’99, (Washington, DC, USA), pp. 162–, IEEE Computer Society, 1999.
45. B. Noble, M. Price, M. Satyanarayanan, and C.-M. U. P. P. D. O. C. SCIENCE., “A programming interface for application-aware adaptation in mobile computing,” *2nd USENIX Symposium on Mobile and Location-Independent Computing*, vol. 8, no. 4, pp. 345–363, 1995.
46. N. Vallina-Rodriguez, C. Efstratiou, G. Xie, and J. Crowcroft, “Enabling opportunistic resources sharing on mobile operating systems: Benefits and challenges,” in *ACM S3 Workshop*, 2011.