# Mobile Computing: Challenges and Opportunities for Autonomy and Feedback

Ole J. Mengshoel
*Carnegie Mellon University*
*Moffett Field, CA 94035*
ole.mengshoel@sv.cmu.edu

Bob Iannucci
*Carnegie Mellon University*
*Moffett Field, CA 94035*
bob@sv.cmu.edu

Abe Ishihara
*Carnegie Mellon University*
*Moffett Field, CA 94035*
abe.ishihara@sv.cmu.edu

## Abstract

Mobile devices have evolved to become computing platforms more similar to desktops and workstations than the cell phones and handsets of yesteryear. Unfortunately, today's mobile infrastructures are mirrors of the wired past. Devices, apps, and networks impact one another, but a systematic approach for allowing them to cooperate is currently missing. We propose an approach that seeks to open key interfaces and to apply feedback and autonomic computing to improve both user experience and mobile system dynamics.

## 1 Background

Mobile computing today represents a discontinuous transformation of the marketplace from an embedded computing perspective to a true platform computing perspective. This single fact, which was in many ways predictable, has re-shaped the mobile industry and has led to a shift in fortunes among mobile equipment makers.

Those who entered the game early focused on delivering voice services. Mobile computing *per se* was the stepchild of voice, the presumed "killer app." A second wave of mobile computing pioneers took the discontinuously different *platform* approach, thinking of computing as primary and communications as a peripheral service–consistent with the way networking interfaces and "modems" were viewed in the desktop world.

The result of this second wave was hundreds of thousands of developers flocking to these new, more platform-like mobile devices. We argue that this *platform thinking* compelled the mobile systems landscape to change, invalidating longstanding design assumptions and bringing new challenges. We observe that the current mobile systems architectural approach – inherited from the past – emphasizes strict layering and separation of function that hides key state information.

It is our contention that the overall system and individual users' experiences can be improved by opening new application programming interfaces (APIs) within the mobile system and utilizing feedback techniques to optimize system performance. A pragmatic approach, which we follow and refine, is to measure and adapt [11]. We imagine creating tools to (i) measure power consumption for computation and communication on-the-fly and (ii) enable *migration* of pieces of apps to or from the cloud, informed by the measurements, using novel feedback and autonomic computing techniques. Our proposed approach, outlined in Section 3, offers a capability to learn the power implications of an app's structure–across many instances concurrently–and to dynamically adapt it at run time.

Our approach builds on previous research including the following. Related to (i), power consumption and battery lifetime for the Openmoko Neo Freerunner smartphone, under different usage scenarios, has been investigated [3]. The power consumption of a G1 Android phone has been recorded and analyzed for a broad range of users and user activities [22]. It has been established that it is sufficient to use the smartphone's battery voltage sensors and knowledge about the battery discharge behavior to accurately estimate power consumption [26]. Related to (ii), the potential for speed-up and energy saving enabled by offloading computation from a smartphone to the cloud, using aspect-oriented programming, is clear [4]. Similar results have been achieved for the mobile Web, based on measured energy usage across different Web sites and different Web page elements [24]. A distinguishing factor of our proposed approach is the use of feedback control theory, which has been applied to computing and computer networks, including control of HTTP servers [1, 6, 7], email servers [15], quality of service assurance [25], internet traffic control [9, 12], and load balancing [10].

In this paper, we attempt to characterize how platform thinking has changed mobile systems, introducing new challenges (Section 2). We select a subset of the chal-

| Challenge | Description |
|---|---|
| Robustness | Wireless characteristics are inherently variable |
| Responsiveness | Growing demand implies growing load |
| Power | Physics imposes hard limits |
| App Development | Distributed computing introduces complexity |

Table 1: Some of the challenges inherent in mobile computing systems.

lenges for which a systems-level approach is particularly applicable and sketch a research plan (Section 3).

## 2 Impact of Platform Thinking

Early mobile systems simply made voice calls. The systems were a co-design of mobile phones, radio access networks (RANs), mobile switching equipment, and associated billing and management subsystems. But with the evolution to platform thinking, new challenges emerged. Powerful computing resources in the phones gave birth to mobile apps, and IP-to-the-phone opened the world of connected mobile computing, with classic mobile networks in the middle. We turn our attention to the key challenges of mobile systems (see Table 1); the state of mobile infrastructure;[1] and the evolution of mobile devices and applications.

### 2.1 Mobile Infrastructure

In traditional wired networks, the physical medium is generally assumed reliable and the devices to which it connects are assumed essentially fixed. But in wireless networks, the physical medium is generally dynamic, variable in reliability, and the devices can and do move, giving rise to the fundamental *robustness* challenge.

With the growth in mobile consumption of streaming media, network *responsiveness* (a function of capacity, load, and engineered-in latencies) has remained a challenge. Just as in wired networks, balancing the competing needs of different traffic flows against fixed resources has revived interest in mechanisms to externally control an otherwise static network (*e.g.*, SDN) as well as policies that can be employed to enforce some notion of rational resource allocation. Real-time resource allocation is a necessity, but current operator practice (*e.g.*, pricing plans for data) treats it as a static problem.

### 2.2 Mobile Devices and Applications

Today, mobile apps rival their desktop counterparts in complexity, visual richness, and real-time interactivity. The competitive nature of mobile app marketplaces is increasingly taxing the computing power of mobile phones, leading to rapid evolution of on-phone computing performance. But this has its physical limits. It is a fact that mobile phone, as we know them, must operate at or below the so-called "three watt limit" [14] – the *power* challenge. Pushing more than three watts through the surface area of a typical phone will make it, quite simply, too hot to handle.

Addressing this issue and others, many mobile apps today are made up of developer's code that provides some unique functionality built on top of one or a number of third-party libraries, each of which is tied to a cloud service (*e.g.*, Dropbox, AdMob, game engines). Mobile apps will, increasingly, be divided between on-device parts and cloud-supplied parts. However, IP packets that travel mobile-to-cloud or mobile-to-mobile have to transit extensive wireless edge and core networks to reach their destinations. Latency is often a problem.

It is the rare app developer who has intuition about how to statically divide an app for power optimization. Depending on how the app is partitioned (between on-device and in-cloud) and the way it is used, the power-cost of computing and the power-cost of communication will change, possibly drastically. While an analytic approach to partitioning and power optimization would be ideal, the inherently unknown nature of the program's input-dependent behavior makes this unrealistic. These facts, coupled with well-known issues of distributed programming make up the *app development* challenge.

## 3 Systems-Level Approach

We propose a new model for mobile systems that takes two steps beyond current systems. First, we propose that app, network and cloud elements expose key bits of state and intent with one another. Second, we propose the creation of a set of feedback and autonomic processes that seek to optimize network behavior and user experience given the information exposed. Space prevents a full exploration of these ideas against the listed challenges, so we explore these two in some depth:

- Apps expose their desire for network *resource allocation* (*e.g.*, bandwidth, maximum latency). The network gathers these and periodically conducts auctions to set prices and priorities. The feedback loop is closed when the apps receive the results of the auction and modify their requests accordingly. Consequently, the network operator maximizes revenue based on the competing requests.

- Apps and networks together participate in *power management*. With instances of the same app running on millions of devices, coupled with explicitly-

[1] We use the term *mobile infrastructure* to mean the RAN, switching, and the IP cloud behind it.

exposed meta-data related to the state of the wireless connections, an opportunity to learn network-dependent power behavior emerges. Correlating power usage with, say, signal strength across many app instances provides the means to adapt program behavior. In a video streaming application, a weak signal could trigger the choice of a codec that would minimize retransmissions, minimizing wasted power.

## 3.1 Resource Allocation

To ensure the health of large-scale mobile systems, it is essential to detect, diagnose, and mitigate performance issues (*e.g.*, an app quickly draining a battery) and faults (*e.g.*, an over-heating battery) as quickly as possible when they occur, and even prognose their likely future occurrence. This is also known as the system health management (SHM) challenge [23]. SHM involves both hardware (sensors, actuators, power supply, electronics, ...) and software (apps, diagnostic and prognostic algorithms, ...). One way to design and implement SHM is by means of probabilistic graphical models, including Bayesian networks (BNs) [16], fed with real-time, precise information about the mobile platform, including the network and the apps. SHM capabilities will need to reside in devices and networks, especially as we move into the age of tens and hundreds of billions of networked devices (i.e., the Internet of Things).

System health management algorithms using BNs [19, 21, 18, 20, 5] have recently been integrated with techniques from control theory [13, 17]. This hybrid feedback control approach promises to improve the self-monitoring and self-adaptive capabilities of mobile systems. To ensure responsive SHM, the controller compares the observed completion time $y(k)$ and desired completion time $r(k)$, and commands the actuator (its output). Specifically, the output of the control algorithm, $u(k)$, determines the maximum number of low-criticality processes running on the computer. Low-criticality processes are suspended or migrated if $v(k) > u(k)$, where $v(k)$ is the current number of low criticality processes. If this happens, the number of active processes on the mobile device is reduced, which again lowers (improves) the output $y(k)$, the responsiveness of the SHM process.

We have applied this approach to state estimation for vehicular electrical power systems, which are more complicated than a smarthpone power supply, but with similar state estimation needs [22, 3, 26]. Our results to date demonstrate how control theory can improve, for a standalone computer, responsiveness under varying computational demands and resources. Improved autonomic and feedback techniques for mobile computing is a natural next step.

## 3.2 Power Management

Power management is perhaps the most pressing issue in mobile app creation and mobility computing. Power usage can be measured across different, concurrent instances of an app, and these measurements can then be correlated with network measurements and models [22, 3, 26]. Machine learning and system identification can be then be done and the resulting combined model used for feedback control for that app. In this case, the setpoint $r(k)$ would be power consumption and the control actions would be to dynamically migrate parts of an app between the device and the cloud in order to meet power consumption requirements. Compared to previous research on offloading [4, 24], we propose to automatically partition a broad class of apps, using feedback control, which is an open research question.

We now discuss in more detail what it means to dynamically partition an app using feedback in order to optimize power consumption for a device. Consider an image processing or augmented reality app. Partitioning of such an app could be done dynamically by selecting between these two options: (i) send image from smartphone to a cloud server and perform feature extraction there or (ii) perform feature extraction on the smartphone, and send features to the server. In short, the dynamic partitioning decision is whether feature extraction is done on the server or on the mobile device. The decision has impact on power consumption, completion time, and data transfer (and thus potentially cost).

The app can be associated with an objective (minimize power consumption), real-time sensor data (actual power consumption), and actuation (pick between options (i) and (ii)[2]) and is therefore suitable for feedback computing. In particular, one can do feedback control similar to reactive Bayesian network computation [13, 17] as discussed above. However, the controller is now triggered by power consumption rather than, or in addition to, completion time.

# 4 Conclusion

The challenges of mobile systems present many new opportunities for autonomic and feedback computing. We have only scratched the surface here, and hope that others will join in developing novel ways of closely integrating control theory with traditional mobile computing, distributed programming, and artificial intelligence techniques. This potentially broader role of control theory in intelligent mobile computing systems parallels other areas of computing in which control theory has already had an impact [8, 2].

[2]More generally, there would be more options per app and potentially many apps that could be actuated

## References


[1] ABDELZAHER, T. F., AND BHATTI, N. Adaptive content delivery for Web server QoS. *Compututer Networking 31* (1999), 1563–1577.

[2] BRUN, Y., DI MARZO SERUGENDO, G., GACEK, C., GIESE, H., KIENLE, H., LITOIU, M., MÜLLER, H., PEZZÈ, M., AND SHAW, M. Engineering self-adaptive systems through feedback loops. In *Software Engineering for Self-Adaptive Systems*, B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, Eds. Springer-Verlag, 2009, pp. 48–70.

[3] CARROLL, A., AND HEISER, G. An analysis of power consumption in a smartphone. In *Proc. of the USENIX Annual Technical Conference* (2010), pp. 271–284.

[4] CHEN, H.-Y., LIN, Y.-H., AND CHENG, C.-M. Coca: Computation offload to clouds using aop. In *Proc. of IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (2012), pp. 466–473.

[5] CHOI, A., DARWICHE, A., ZHENG, L., AND MENGSHOEL, O. J. A tutorial on Bayesian networks for system health management. In *Data Mining in Systems Health Management: Detection, Diagnostics, and Prognostics*, A. Srivastava and J. Han, Eds. Chapman and Hall/CRC Press, 2011.

[6] DIAO, Y., GANDHI, N., HELLERSTEIN, J. L., PAREKH, S., AND TILBURY, D. M. Using mimo feedback control to enforce policies for interrelated metrics with application to the Apache Web server. In *Proc. of Network Operations and Management Symposium (NOMS-02)* (Florence, Italy, 2002), pp. 219–234.

[7] DIAO, Y., HELLERSTEIN, J. L., AND PAREKH, S. Optimizing quality of service using fuzzy control. In *Proc. of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management: Management Technologies for E-Commerce and E-Business Applications* (London, UK, 2002), DSOM'02, pp. 42–53.

[8] HELLERSTEIN, J., DIAO, Y., PAREKH, S., AND TILBURY, D. M. *Feedback Control of Computing Systems*. Wiley, 2004.

[9] HOLLOT, C., MISRA, V., TOWSLEY, D., AND GONG, W. On designing improved controllers for AQM routers supporting TCP flows. In *Proc. of IEEE INFOCOM* (2000), pp. 1726–1734.

[10] KAMEDA, H., FATHY, E.-Z., RYU, I., AND LI, J. A performance comparison of dynamic vs. static load balancing policies in a mainframe-personal computer network model. In *Proc. of the 39th IEEE Conference on Decision and Control* (2000), vol. 2, pp. 1415–1420.

[11] KREMER, U., HICKS, J., AND REHG, J. A compilation framework for power and energy management on mobile computers. In *Proc. of the 14th international conference on Languages and compilers for parallel computing* (2003), LCPC'01, pp. 115–131.

[12] MAGHSOUDLOU, A. R., BARZAMINI, R., SOLEIMANPOUR, S., AND JOUZDANI, J. Neuro fuzzy model predictive control of AQM networks supporting TCP flows. *Proc. Ninth ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel Distributed Computing* (2008), 226–230.

[13] MENGSHOEL, O. J., ISHIHARA, A., AND REED, E. Reactive Bayesian network computation using feedback control: An empirical study. In *Proc. of BMAW-12* (Catalina Island, CA, August 2012).

[14] NEUVO, Y. Cellular phones as embedded systems. In *Proc. of IEEE International Solid-State Circuits Conference* (2004), pp. 32–37.

[15] PAREKH, S., GANDHI, N., HELLERSTEIN, J., TILBURY, D., JAYRAM, T., AND BIGUS, J. Using control theory to achieve service level objectives in performance management. *Real-Time Systems 23*, 1 (2002), 841–854.

[16] PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[17] REED, E., ISHIHARA, A., AND MENGSHOEL, O. J. Adaptive control of Bayesian network computation. In *Proc. of 2012 - 5th International Symposium on Resilient Control Systems* (Salt Lake City, UT, August 2012).

[18] RICKS, B. W., HARRISON, C., AND MENGSHOEL, O. J. Integrating probabilistic reasoning and statistical quality control techniques for fault diagnosis in hybrid domains. In *Proc. of the Annual Conference of the PHM Society 2011 (PHM-11)* (Montreal, Canada, 2011).

[19] RICKS, B. W., AND MENGSHOEL, O. J. Methods for probabilistic fault diagnosis: An electrical

power system case study. In *Proc. of Annual Conference of the PHM Society, 2009 (PHM-09)* (San Diego, CA, 2009).

[20] SCHUMANN, J., MENGSHOEL, O. J., AND MBAYA, T. Integrated software and sensor health management for small spacecraft. In *Proc. of IEEE International Conference on Space Mission Challenges for Information Technology* (Palo Alto, CA, 2011), pp. 77–84.

[21] SCHUMANN, J., MENGSHOEL, O. J., SRIVASTAVA, A. N., AND DARWICHE, A. Towards software health management with Bayesian networks. In *Proceedings of the FSE/SDP workshop on Future of software engineering research* (Santa Fe, New Mexico, 2010), FoSER '10, ACM, pp. 331–336.

[22] SHYE, A., SCHOLBROCK, B., AND MEMIK, G. Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures. In *Proc. of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture* (2009), pp. 168–178.

[23] SRIVASTAVA, A., AND HAN, J., Eds. *Data Mining in Systems Health Management: Detection, Diagnostics, and Prognostics*. Chapman and Hall/CRC Press, 2011.

[24] THIAGARAJAN, N., AGGARWAL, G., NICOARA, A., BONEH, D., AND SINGH, J. P. Who killed my battery?: analyzing mobile browser energy consumption. In *Proc. of the 21st International Conference on the World Wide Web* (2012), pp. 41–50.

[25] XU, C.-Z., LIU, B., AND WEI, J. Model predictive feedback control for QoS assurance in webservers. *Computer 41* (March 2008), 66–72.

[26] ZHANG, L., TIWANA, B., QIAN, Z., WANG, Z., DICK, R. P., MAO, Z. M., AND YANG, L. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proc. of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis* (2010), pp. 105–114.