Poster: Harnessing Spectrum Awareness to Enhance Mobile Computing

S. Eman Mahmoodi
Stevens Institute of Technology
Hoboken, NJ 07030
smahmood@stevens.edu

K. P. Subbalakshmi
Stevens Institute of Technology
Hoboken, NJ 07030
ksubbala@stevens.edu

R. N. Uma
North Carolina Central University
Durham, NC 27707
ruma@nccu.edu

ABSTRACT

The well documented growth in mobile traffic is mainly driven by increasingly sophisticated smart phone applications. Simultaneously, user preference for lighter phones has resulted in more battery power constrained hand-helds that offload computations to resource intensive cloud. This second trend exacerbates the bandwidth crunch often experienced over wireless networks. *Our idea (joint cognitive offloading and scheduling) is to use dynamic spectrum access and management concepts from wireless networking to effect computation offloading and scheduling solutions that achieves near optimal trade-offs between the mobile device and wireless resources.* We use all radio available interfaces (e.g. WiFi, LTE) in multi-RAT enabled devices to schedule appropriate components of the application to run either on the mobile device or on the resource-rich cloud, while staying adaptive to the conditions of the wireless network.

1. INTRODUCTION

Fig. 1 shows an overview of a scenario where our proposed approach can be applied. In this figure, N-component (features including AR, mobile healthcare, public safety, IT services, etc) mobile applications are being run on mobile devices within a macro-cell coverage area where several wireless networks (WiFi hotspots, mobile cloudlets, femtocells etc) coexist. Consider mobile devices A, B, and C that use the MEC cloud server in the femtocell and the nearby WiFi hot spot (connected to the EC2 web-based cloud) for cognitive offloading. Therefore, there are three computation resources for running the mobile application (mobile, MEC, EC2). The proposed cognitive Offloader and Scheduler (COS) decides which components of the application should be offloaded to the remote cloud resources (MEC, EC2) and which should run locally. Simultaneously, it decides which radio interfaces (femtocell, WiFi hot spot) must be used in the associated data transfers and what percentage of the data should be communicated through each interface. We also see that mobile users D and E use the MEC server of femtocell and the cellular radio for cognitive offloading. The single-radio enabled wearable device (F) uses MEC server, and the mobile user G uses a nearby WiFi and the cellular daio for using the proposed COS scheme to run the 10-component multi-media app.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MobiCom'16 October 03-07, 2016, New York City, NY, USA

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4226-1/16/10.
DOI: http://dx.doi.org/10.1145/2973750.2985277

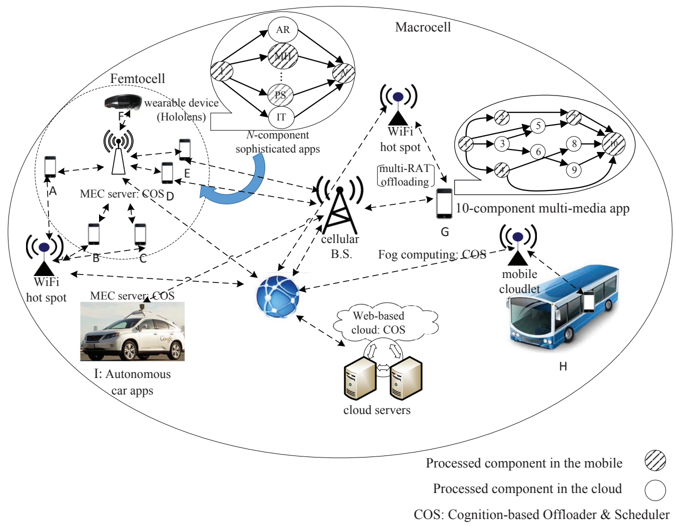


Figure 1: The Future: joint cognitive offloading and scheduling to achieve dynamic mobile computing. Using this scheme all mobile users could run sophisticated complex mobile applications via multiple computational resources through multiple radio interfaces. This scheme is implemented on (1) Mobile Edge Computing (MEC) servers through mobile cloudlets and gateways of fog computing protocols , and (2) Internet-based cloud servers (e.g. Amazon EC2, NSFCloud). This spectrum-aware computation offloading is able to aggregate bandwidths from appropriate wireless backhaul networks (e.g. Verizon, AT&T).

2. JOINT COGNITIVE OFFLOADING AND SCHEDULING

We propose the idea of a joint cognitive offloading and scheduling, where the computational offloader not only decides which components of a complex application should be offloaded to the remote cloud resources (a combination of MECs, cloud servers) and which should run locally, but also which radio interfaces must be used in the associated data transfers and what percentage of the data should be communicated through each interface. Cognitive use of all the wireless interfaces *simultaneously* leads to a higher throughput of the networks as well as efficient use of wireless resources. Moreover, since the idea accounts for the component dependency graphs of the application, it can also allow for parallel execution of appropriate components thereby reducing application runtime and enhancing user experience. Note that although this scheme is primarily designed for multi-RAT enabled devices, it provides the best solution for legacy single radio enabled devices by enabling partial cloud offloading and wireless aware scheduling.

We propose and maximize a multi-objective net utility that trades-off resources saved in the mobile device (battery power, memory, CPU) with the resources consumed (communication energy, delay). The constraints that should be monitored include deadlines of application runtime, precedence order of components of an application, scheduling order, series/parallel processing requirements of computational resources, and stability of the multi-RAT data queue. This scheme provides a wireless-aware scheduling service, which allows for more degrees of freedom in the solution by moving away from a compiler pre-determined scheduling order for the components towards a more wireless-aware scheduling order. By being component dependency graph (CDG) aware, the solution is also highly parallelizable, thereby reducing application runtimes. Moreover, this is a time-adaptive scheme for realtime services such that decision making is dynamically adapted with random instantaneous variations of wireless parameters including data rate, delay, and queue backlog of transmission buffers.

The proposed cognition-based offloader and scheduler (COS) comes with variants of strategies dependent on the use cases as follows: (*i*) Realtime optimal solution, which provides the maximum net utility at the cost of high computational complexity. (*ii*) Realtime heuristic solution [1], which gives a sub-optimal solution (only 7% lower net utility compared to the optimal solution) but with the advantage of instantaneous decision making. This solution is application to uses cases where high speed, real-time adaptivity is desirable, such as Google self driving cars. (*iii*) Offline optimal solution determines the maximum net utility using the average wireless parameters. Although this service does not adapt with dynamic changes of mobile networks, it gives the best solution in static environments and is significantly faster. An example use case could be non-critical mobile healthcare application that monitors average status of patient health over a day.

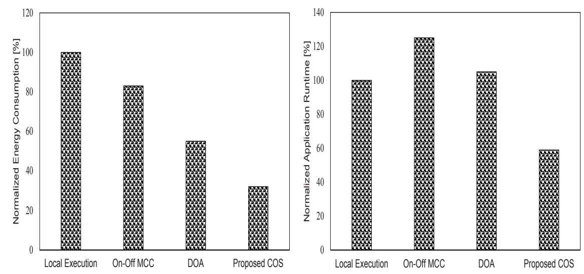
3. EXPERIMENTAL SETUP

The proposed schemes were analyzed using real data collected for complex applications running on HTC Vivid smartphone with a 1.2GHz dual-core processor associated with the two cloud servers: NSFCloud [1] (Figs 2(a,b,c)) and Amazon EC2 (Fig. 2d), and two radios: WiFi and LTE (the proposed schemes are developed for general K radio interfaces). Data was gathered for wireless networks in both the indoor and outdoor environments. To check for scalability interms of application complexity, we tested the schemes with a variety of CDG dependencies using apps such as a face recognition app [2] and a video navigation app (features including: graphics, video processing, object recognition, clustering points). The uplink and downlink data rates and delays for WiFi and LTE (indoor and outdoor) were obtained using the Android FTP tool.

Our proposed work (COS) is compared with several other schemes including: (*i*) mobile-only execution where all the application components are processed in the mobile device (local execution); (*ii*) On-Off remote mobile cloud computing (On-Off MCC) where all components must be offloaded and at each time epoch, the offload is scheduled on the best available wireless interface (On-Off); (*iii*) remote MCC using single radio interfaces (WiFi and LTE); and (*iv*) the dynamic offloading algorithm (DOA) proposed in [2].

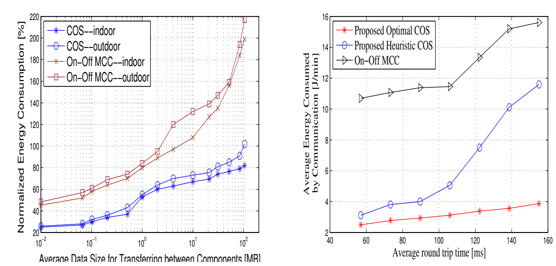
4. RESULTS

All the simulations in Fig. 2 are for realtime (online) COS scheme. Fig. 2a shows the comparison of normalized (normalized w.r.t. the energy consumed by mobile-only execution) energy consumed by the mobile. We observe that COS consumes 68%, 51%, and 23% less energy in comparison to the schemes using local execution, On-Off MCC, and DOA, respectively. In Fig. 2b, runtime values of the face-recognition application (normalized to the scheme with local execution) are compared. The COS enables the application to be processed 41%, 66%, and 46% faster compared to the schemes using local execution, On-Off MCC, and DOA, respectively. Fig. 2c plots the total normalized (w.r.t energy consumed to run the application fully on the mobile) versus the average offload data size in indoor and outdoor environments. As expected, the energy consumed increases with data size, for all schemes and environments. Mobile device consumes less energy in the indoor environment in comparison to the outdoor. The proposed COS outperforms On-Off MCC 48.5% on average. Note that MCC On-Off consumes higher energy even in comparison to the mobile-only execution in higher ranges of data sizes. This happens for data sizes exceeding 10MB. Fig. 2d shows the average energy consumed for offloading versus the average RTT. We can see that while latency increases, more energy is consumed for communication.



(a) Total energy consumed (normalized to the energy consumed by mobile-only execution) for the proposed and classical schemes using a face recognition application.

(b) Total application runtime (normalized to application runtime in the mobile) for the proposed and classical schemes using a face recognition application.



(c) Normalized total energy consumed by the mobile device versus average data size required for transferring between components using our video navigation application.

(d) Average energy consumption versus average round trip time (runtime= 1330 ms, delay threshold for offloading= 550 ms) using our video navigation application.

Figure 2: Results in the online scenario.

5. REFERENCES

[1] S. E. Mahmoodi and K. P. Subbalakshmi, "A time-adaptive heuristic for cognitive mobile cloud offloading for multi-radio enabled wireless devices," *IEEE Transactions on Cognitive Communication and Networking*, early access, July 2016.

[2] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, Jun. 2012.

[1] www.chameleoncloud.org/nsf-cloud-workshop/.

[2] http://darnok.org/programming/face-recognition/