All Opportunities Are Not Equal: Enabling Energy Efficient App Syncs In Diverse Networks

Priyanka Tembey, Saumitra Das, Dilma Da Silva, Vrajesh Bhavsar
Qualcomm Research Silicon Valley

Abstract

Mobile apps connect with counterpart cloud services and pursue data transfer over wireless networks. The amount of mobile data transferred will only increase as app data needs and usage expand. Energy expenditure during periods of data transfer constitutes a significant portion of a mobile's battery usage. At the same time, due to the evolving nature of wireless technologies, there is a proliferation of networks (such as public Wifi hotspots, access points, and cellular technologies) that exhibit diverse energy and performance characteristics. As a user moves around (even within the same logical network) the hardware serving data transfer varies, often offering opportunities for data transfer with distinct bandwidth and latency capabilities. Apps and data services in smartphones however are oblivious to this diversity and the energy impact of using one opportunity vs. others.

In this paper, we first present a study of Wifi characteristics in two domains, shopping malls and within an enterprise campus, demonstrating the fine-grained diversity of network opportunities in two commonplace scenarios. Next, we describe a system-level framework and set of interfaces that enable energy efficient app syncs by leveraging the right opportunities. Preliminary results using our approach show up to three times lower energy costs for popular mobile apps using media uploads.

1 Introduction

Many popular mobile apps send and receive data by connecting with cloud services. With the arrival of Internet of Things (IoT) apps [1, 11], mobile data traffic will only increase in the future. Mobile apps use cellular and/or Wifi networks to sync data with cloud services. Due to the constant evolution of wireless standards, network conditions vary: switching between 3G/4G/HSPA+ is fairly commonplace in the cellular world, while switching between 802.11ac/b/g/n is prevalent in public Wifi mobility domains. These networks are heterogeneous in the bandwidth and latency they offer, hence presenting different data transfer opportunities that differ in their energy impact on mobile devices.

Many data transfer requests issued by mobile apps are delay-tolerant, i.e., the app behavior does not depend on data transfer requests being carried out immediately. Examples of apps that issue delay-tolerant data transfer requests include news feed apps like NYTimes; popular social media apps such as Twitter; Facebook and Instagram; and media upload apps such as Google Auto Photo Backup. Wearables and IoT apps also exhibit delay-tolerance in periodically syncing their data with cloud services via smartphones [6, 1] and/or smart hubs [11].

Data movement is central to a diversity of mobile apps, and it is responsible for a significant fraction of the phones energy expenditure. However, there is no system-level support in smartphones to minimize energy costs for delay-tolerant data transfers across a variety of WiFi and cellular technologies. In this paper, we present a generic system-level framework which comprises (1) a learning engine that characterizes 'opportunities' for data transfer personalized to a user's mobility pattern, (2) set of APIs that convey knowledge of opportunities to data transfer schedulers, and (3) data transfer schedulers that leverage app delay tolerance to choose the best time to carry out the transfer, thereby achieving energy savings.

In this paper, we focus primarily on opportunities observed within networks with single Wifi SSiDs consisting of multiple access points (APs). Common scenarios include Wifi networks in college campuses, enterprise, and malls. The variation in opportunities observed in these networks result from the diversity in AP WiFi bands (ac/g/b/n) and their relative signal strengths.

Previous work has demonstrated the impact of signal strength on mobile energy consumption [5]. Rahmati and Zhong [9] propose decision models based on users' personal navigation to opportunistically offload cellular data to Wifi networks for energy efficiency, while other related work [2, 7, 3] adds prediction of Wifi opportunities and scheduling support to offload cellular data for energy efficiency. Breadcrumbs [8] adds predictive forecasting capabilities for users' Wifi mobility, while Bartendr [10] tracks and predicts users' cellular mobility so as to schedule data transfers using better signal strengths. Our approach builds upon prediction models of previ-

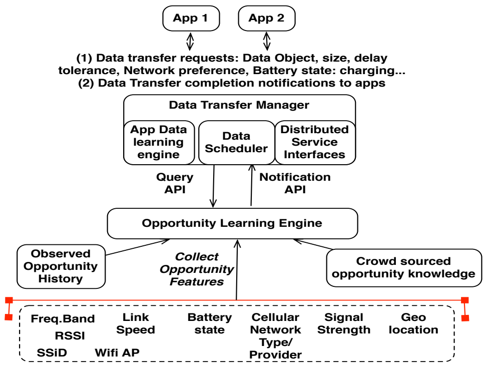


Figure 1: System Architecture

ous work [8, 2], while presenting a generic and extensible system framework and interfaces to enable all delay-tolerant apps (including IoT apps) to leverage network opportunities. We also focus on a usage scenario not explored so far in the literature: multi-AP SSiD WiFi networks. We show that this scenario can present fine-grained variation of data transfer opportunities and that by properly scheduling data transfers, mobile apps can achieve substantial energy savings.

2 Architecture

Figure 1 shows our system architecture. Next we describe the system components that, together, enable mobile apps to achieve more energy efficient data transfers with their cloud services.

2.1 Opportunity Learning Engine

As a user's device navigates through wireless networks, the Learning Engine identifies personalized opportunities for the user, and interfaces with higher-level data schedulers in the system to notify them of 'good' data transfer opportunities for pursuing data transfer. It performs the following functions.

1. Opportunity Feature collection. We define an opportunity as a set of network features that characterize conditions in which data transfer occurs. In previous work [5, 8, 9], signal strength (i.e., RSSI) is used as a feature to discern varying network conditions. We propose a finer grained set of features to define an opportunity.

For cellular networks, the tuple *(cellular network type, provider, signal strength, geo-location)* is used to discern differences in bandwidth and latency seen across 3G/LTE networks, connected base-stations and their providers, hence presenting different opportunities. For Wifi networks, we use the tuple *(SSid, AP MAC, Link Speed, Freq.band, RSSI)*. SSids, such as of home, user's work Wifi, public hotspots in cafes and airports can help generalize across backhaul providers (e.g., Comcast, AT&T). Connecting with multiple access points (APs) supporting different Wifi standards (802.11 a/b/g/n) is fairly common within a single SSiD domain (such as in college campuses and enterprises). Depending on their wireless standard, APs support connections in 2.4 and/or 5GHz range, which influences observed link speed at various signal strengths, hence resulting in different opportunities. In ongoing work, we are exploring load-based opportunities [4], low load being a better opportunity. Finally, the battery state of the device also influences opportunities: if the device is charging, all opportunities will be weighted equally at zero energy cost to the device. Features are collected periodically when the device is on.

2. Model generation In addition to collection of network features, our system also monitors throughput and latency seen in different network conditions. For this purpose, we use active probing as used in previous work [9, 8]. Our approach builds on the prediction models of previous work [8] by using supervised learning techniques to classify opportunities (i.e., network feature-sets). Good opportunities with low latency and high throughput have a higher 'opportunity score' than bad ones with high latency and low throughput. Our system also distinguishes between upload and download opportunities. Monitoring overhead will further reduce in time as the learning engine derives more knowledge from historical state: the learning engine may learn the maximum achievable throughput and RSSI in a user's home Wifi network over time, so it can avoid recalibrating opportunity scores and enable data transfer when 'best' opportunities in a network are encountered. In comparison with previous work [8, 10], our system also aims to explore and predict longer-term opportunities: in terms of minutes and hours. Example scenarios where delaying data-heavy transfers over longer time-scales is beneficial may arise if users tend to connect with particularly bad APs within office and campus networks, however they are likely to encounter better network opportunities minutes or hours in the future, known via their previous network connection history.

3. Crowd sourced opportunity knowledge The learning engine will run on smartphones of multiple users, and periodically upload opportunity scores to a secure cloud service. For instance, opportunity scores of public Wifi networks are learned over time, and new users retrieve this information to skip their own learning phase, hence reducing system overhead in learning new opportunities.

4. Notification API to higher level data services Opportunity scores are exported to higher level managers via a notification and query API. The learning engine sends notifications in the form of a message with the parameters: *(Opportunity score, Rank, Time to last, Confidence percent)*. For instance, the learning engine communicates the 'best' opportunity (Rank: 1) seen for a network, along with its estimated duration and prediction confidence. Such notifications may be leveraged by higher level managers to schedule backed up data transfer or prefetch requests. Higher level managers may also query the learning engine seeking (a) current opportunity scores, (b) opportunity score 'x' steps in the future [8], time to last and how it relatively ranks compared to previously observed opportunities in the network. We envision richer query APIs seeking responses to (a) how further away in time or distance is the best opportunity for the current network? or (b) will there be a better opportunity in a finite 'y' time interval?

2.2 Data Transfer Manager

The Data Transfer Manager is responsible for maintaining a schedule of data transfers to be performed that abide by app delay tolerance parameters. It also interfaces with the opportunity learning engine and, possibly, with data managers on other nearby devices.

1. Learning app data requirements. The Data Transfer Manager collects and maintains state regarding app data objects and their delay tolerance. This can be done transparently using binary analysis and instrumentation (e.g., photo object uploads can be detected and always be delayed), or by explicitly adopting an API that allows applications to specify their delay tolerance.

2. Scheduling data transfer. Our system maintains data request scheduling queues for mobile apps, and interfaces with the kernel network stack to schedule data transfer. Decision algorithms consider the opportunity scores and their lasting times provided by the opportunity learning engine.

3. Distributed service interfaces. Gateways may implement smart services for IoT things or smartphone apps in the future. Examples include caching, private storage, and prefetching app content. While the opportunity learning engine is only aware of network level opportunities, we envision 'service level opportunities' on nearby smart devices that can be leveraged by smartphone data transfer managers for more energy-efficient data transfer. For example, we implemented an Amazon S3 service instance on a gateway and found photo uploads from smartphones to consume 40% less energy and time when uploading to the gateway S3 instance vs. Amazon's S3 service in the cloud.

Data Transfer Managers for IoT. Wearable devices such as Fitbit Flex and smart-watches periodically sync with cloud apps via smartphones or smart hubs. Most of these sync requests (e.g., step counter syncs) are delay tolerant. In order to improve their energy efficiency on battery-constrained IoT devices, we envision the opportunity learning engine and data transfer manager components as extensible to IoT devices, where they will identify good opportunities to schedule syncs to intermediate devices such as smartphones. Our system architecture can hence be extended and generalized for other devices.

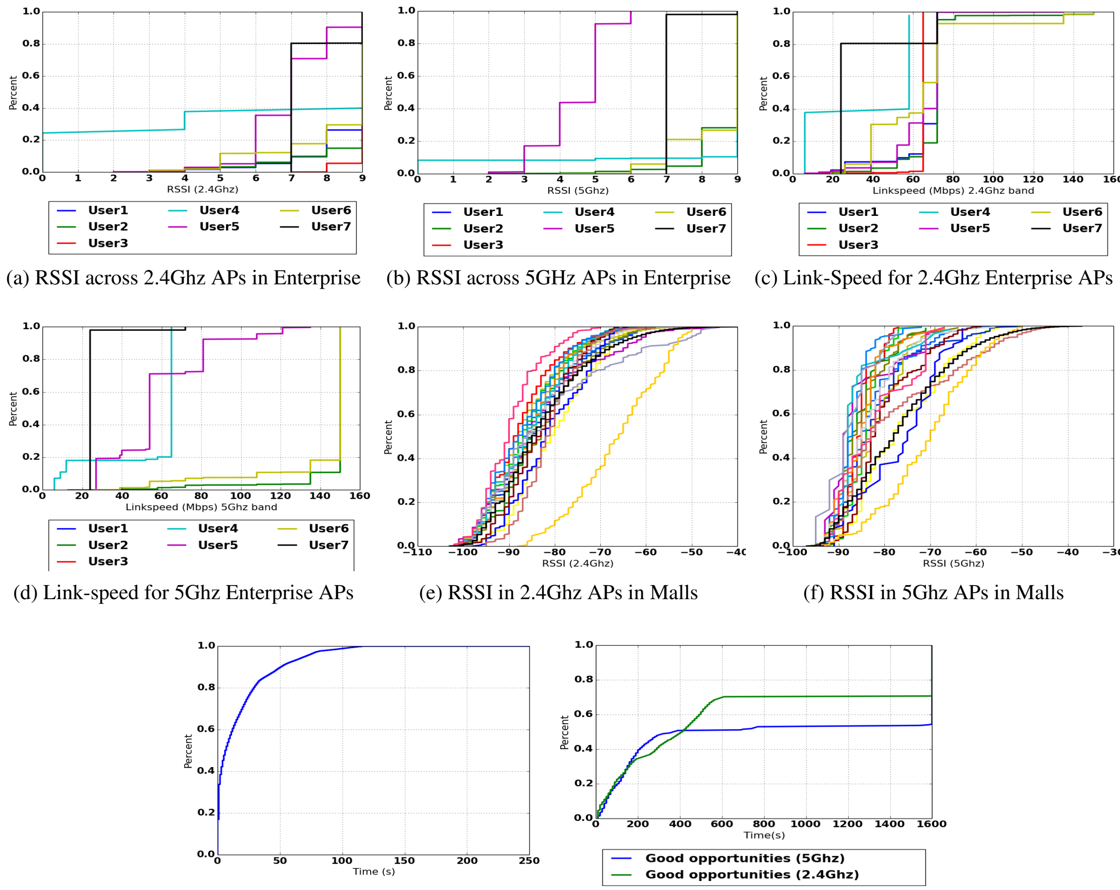
2.3 Implementation in Android

Next we describe how our system can be integrated in an existing smartphone ecosystem. Android uses a sync service comprising sync request queues, a sync scheduler and sync APIs. We propose an extension of the Android Sync API that incorporates delay tolerance. The data transfer manager functions can be implemented in the Android Sync scheduler with appropriate listener hooks for broadcast notifications from the opportunity learning engine service. Similarly, the scheduler can communicate queries via message passing to the learning engine. The Sync service can also be extended to sync IoT and wearable device data.

3 Do opportunities vary?

For a common usage scenario of multi-AP SSiD WiFi networks in enterprises and shopping malls, we show the presence of fine-grained variation of data transfer opportunities (due to variation in RSSI, frequency bands and link-speeds as described in Section 2.1). For the enterprise WiFi case, we installed a logger app that periodically collected these network features on our colleagues' smartphones for a week, as they walked around in the office. We used the same logger app while walking around a large scale shopping mall to collect its network features.

Opportunities vary considerably. Figures 2a and 2b show CDFs for RSSI levels encountered for 2.4 and 5Ghz APs in Enterprise Wifi setting for 7 users. Our first observation is that APs in both frequency bands (2.4 and 5Ghz) are commonly seen in multi-AP SSiD networks. For instance, User-6's smartphone (HTC One) connects with 39 different APs in the 2.4Ghz band, and 37 different APs in the 5Ghz band. Some older smartphones (such as User-1's Samsung S3) may not support 5Ghz Wifi connections, and will miss those opportunities. Second, there is considerable variation in signal strengths even within a frequency band. For instance, for User-6, 10% of RSSI levels are < 7 across APs in the 2.4Ghz band, while for User-4, 40% of the RSSI levels are < 4 signifying that 40% of opportunities are 'bad'. Figures 2c and 2d show how RSSI and frequency bands together affect observed theoretical link-speeds. Most of the observed link-speeds (90% of all) in the 2.4Ghz band are capped at 72Mbps for the best RSSI level of 9, while they are bumped up to 150Mbps for APs in the 5Ghz band. For lower RSSIs (< 3), the link-speeds can be as low as 6Mbps for User-4 in Figure 2c. Finally these findings extend to larger and denser multi-AP SSiD networks such as shopping malls. In one trajectory, a user encountered 562 APs in the 2.4Ghz band, and 236 APs in the



(a) RSSI across 2.4Ghz APs in Enterprise (b) RSSI across 5GHz APs in Enterprise (c) Link-Speed for 2.4Ghz Enterprise APs

(d) Link-speed for 5Ghz Enterprise APs (e) RSSI in 2.4Ghz APs in Malls (f) RSSI in 5Ghz APs in Malls

(g) Arrivals of good opportunities in Malls (h) Arrivals of good opportunities in Enterprise

Figure 2: Opportunities in enterprise and large-scale shopping mall wifi networks

5Ghz band. As observed in Figures 2e and 2f, there is considerable variation from ‘bad’ to ‘good’ RSSI levels across APs in *both* bands for all 20 user trajectories.

Waiting for good opportunities is feasible. In networks with good and bad opportunities, how long would a data transfer have to be delayed to use a good opportunity instead of a currently bad one? We characterized observed opportunities as ‘good’ and ‘bad’ opportunities using a simple classification: Opportunities in the 2.4Ghz band with RSSI levels < 3 are bad opportunities, while those in the 5Ghz band with RSSI levels $>=7$ are good opportunities (Note that opportunities with RSSI $>= 7$ in the 2.4Ghz band are also good opportunities, however the ones in 5Ghz band have better throughput capability). Figures 2g and 2h show the CDFs of arrival times of the first good opportunity after the current bad one in the mall and the enterprise respectively. In a denser network where people move around often, such as the mall, 90% of the time good opportunities are encountered within a minute. In an enterprise campus, the delays may be longer (50% of the time, good opportunities are observed in under 10 minutes), as the use-case is more sedentary and the network is less dense in APs. We relax the definition of ‘good’ opportunity to also include 2.4Ghz APs with RSSI $>= 7$ in the enterprise setting. This bumps up the probability of seeing a good opportunity under 10 minutes to 70%. However, we envision that for more active users or settings (e.g., in manufacturing work spaces), these delays will be further bounded.

Systems that enable apps to leverage delay-tolerance for energy efficiency will see more observable benefit in networks with a mixture of good and bad opportunities. We find our study in opportunities encouraging in that (a) Opportunities vary and at times 30% of them may be ‘bad’ while 30% may be ‘good’ as well; (b) for our experimental scenarios, transition times from bad to good opportunities are feasible enough to allow data transfer to be delayed towards periods of good opportunity.

4 Do opportunities matter?

In the next set of experiments, we first use a simple active-probing-based benchmark to demonstrate how our system tracks data transfer throughput in our enterprise

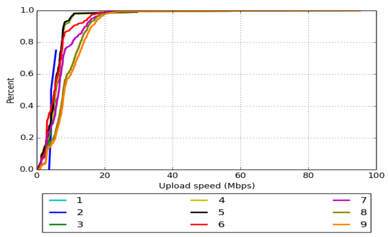


Figure 3: Upload throughput

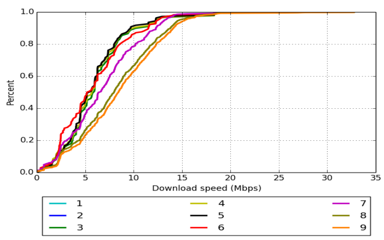


Figure 4: Download throughput

network. Next, the knowledge gained from modeling the enterprise network's throughput is used to classify opportunities as 'good' and 'bad' ones by defining thresholds for signal strengths and frequency bands. This is then used to choose the best time to upload media (photos and videos sized 170MB in total) to an Amazon Cloud service instance, thereby achieving energy savings.

Throughput benchmark. We develop a mobile app that periodically downloads and uploads a chunk of data to a server machine in our network. We deploy it on a user's Nexus 5 smartphone (User 5 in Figure 2a) and track throughput observed as the user roams around in the office, while connecting to APs in different frequency bands(ac/n/g) over varying signal strengths. Figures 3 and 4 respectively show the CDFs of upload and download throughput results. We can observe from the figures that a greater fraction of both upload and download throughput measurements are higher for good RSSI levels (8 and 9) vs. weak RSSI levels (1/2/3). For example, 90% of upload throughput values are lower than 8Mbps for RSSIs < 4, compared with 20Mbps for RSSIs $>= 8$. The smartphone connects with both 2.4Ghz and 5Ghz APs in this experiment, but we do not present the distinction amongst signal strengths based on bands for lack of space. We observe higher throughputs for same RSSI levels in 5Ghz APs vs. 2.4Ghz ones.

Media upload app. Next, by using RSSI thresholds to classify opportunities as 'good' (RSSI $>= 7$) and 'bad' (RSSIs < 4), we delay media upload requests to a cloud service from an app towards periods of good opportunities in our enterprise network. We also track the AP MAC address used for the data transfer and further use our knowledge of mapping AP MAC addresses to their

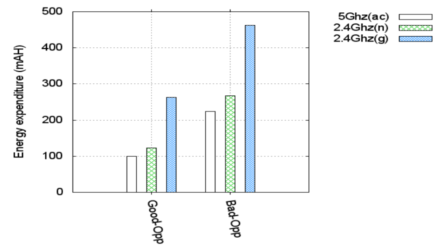


Figure 5: Energy expenditure for Media upload to Amazon S3 using different opportunities

types (ac/n/g) to distinguish between opportunities (Note that this can also be ascertained dynamically using frequency band and link-speed features collected by our logger app). We observe in Figure 5 that delaying media uploads from the worst opportunity in this experiment (over 2.4Ghz(g) with bad RSSI) to the best opportunity (over 5Ghz(ac) with good RSSI), leads to 300% energy savings. These are over and above the energy savings achieved using purely RSSI level opportunity distinction (over 2.4Ghz(g) with good RSSI) by 1̃50%. Choosing a 5Ghz(ac) AP over a 2.4Ghz(n) AP leads to energy benefits of 30%.

5 Conclusions and Future Work

In this paper, we focus on a commonly observed usage scenario, multi-AP SSiD networks, and demonstrate the fine-grained data transfer opportunities in two real networks: in a shopping mall and in an enterprise setting. In order to leverage such opportunities, we propose a generic architecture and set of interfaces in smartphones to minimize energy costs of delay-tolerant data transfers. Preliminary measurements using our approach show 30-300% energy savings when uploading media to a cloud service. Ongoing work extends the notion of opportunities to both multi-SSiD WiFi and cellular networks.

References

[1] APPLE. "https://www.apple.com/ios/ios8/health/", 2014. "Apple HealthKit".

[2] BALASUBRAMANIAN, A., MAHAJAN, R., AND VENKATARAMANI, A. Augmenting Mobile 3G Using WiFi. In *MobiSys* (2010).

[3] BILGIR YETIM, O., AND MARTONOSI, M. Adaptive Usage of Cellular and WiFi Bandwidth: An Optimal Scheduling Formulation. In *CHANTS* (2012).

[4] CHAKRABORTY, A., NAVDA, V., ET AL. Coordinating Cellular Background Transfers Using Loadsense. In *MobiCom* (2013).

[5] DING, N., WAGNER, D., ET AL. Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain. In *SIGMETRICS* (2013).

[6] FITBIT. "https://www.fitbit.com/flex", 2012. "Fitbit".

[7] LEE, K., LEE, J., YI, Y., RHEE, I., AND CHONG, S. Mobile Data Offloading: How Much Can WiFi Deliver? In *Co-Next* (2010).

[8] NICHOLSON, A. J., AND NOBLE, B. D. BreadCrumbs: Forecasting Mobile Connectivity. In *MobiCom* (2008).

[9] RAHMATI, A., AND ZHONG, L. Context-for-wireless: Context-sensitive Energy-efficient Wireless Data Transfer. In *MobiSys* (2007).

[10] SCHULMAN, A., NAVDA, V., , ET AL. Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling. In *MobiCom* (2010).

[11] SMARTTHINGS. "http://blog.smartthings.com/", 2013. "SmartThings".