Stochastic optimization for Conserving Energy in Smart Phone Applications - A Survey

M. Ramu, V. Indurani
Asst. Prof., SVCET

Abstract **- Many mobile applications require frequent wireless transmissions between the content provider and mobile devices, consuming much energy in mobile devices. Motivated by the popularity of prefetch-friendly or delay-tolerant apps (e.g., social networking, app updates, cloud storage), are enabled by the ability to capture videos on a smartphone and to have these videos uploaded to an Internet connected server. Smartphones have multiple wireless interfaces – 3G/EDGE and WiFi – for data transfer, but there is considerable variability in the availability and achievable data transfer rate for these networks. Moreover, the energy costs for transmitting a given amount of data on these wireless interfaces can differ by an order of magnitude. On the other hand, many of these applications are often naturally delay-tolerant, so that it is possible to delay data transfers until a lower-energy WiFi connection becomes available. In this paper, we used the optimal online algorithm for this *energy-delay tradeoff* using the Lyapunov optimization framework algorithm, called SALSA, can automatically adapt to channel conditions and requires only local information to decide whether and when to defer a transmission. The SALSA can be tuned to achieve a broad spectrum of energy-delay tradeoffs, is closer to an empirically-determined optimal than any of the alternatives we compare it to, and, can save 10-40% of battery capacity for some workloads.**

Index Terms- **Mobile cloud computing, energy efficiency, transmission management, Interface Selection, Smartphone, Lyapunov Optimization**

I. INTRODUCTION

With an increasing popularity of mobile devices in recent years, users are gradually shifting their preferences from traditional cell phones and laptops to smartphones and tablets. As indicated by Cisco [3], traffic from mobile devices is anticipated to exceed that of wired devices by 2014 and to account for 61 percent of the entire IP traffic by 2016. The prosperity of mobile markets is largely driven by rich-media applications—over two thirds of the global mobile traffic will be rich-media content including video, image and audio by 2017 [3]. The statistics from Flurry [4] further confirm that the time users spend on mobile applications has surpassed that of web browsing on both desktop and mobile devices since 2011, while users spend 80 percent of their time on rich-media apps such as social networking, gaming, news feed and videos.

To meet the demand for ubiquitous access to rich-media content, many developers have started to leverage cloud computing to overcome resource constraints on mobile devices. Mobile cloud computing is emerging as a new computing paradigm which has fostered a wide range of exciting rich-media applications (e.g., Instagram, Viddy, Siri). However, these mobile-cloud applications require heavy data transmissions, which consume a significant portion of the mobile device battery through the use of its network interface [5]. The transmission burden is further exacerbated by cloud-based data backup and mobile-cloud storage services such as Dropbox, which stores and synchronizes mobile data in Amazon S3 storage system. In fact, according to Cisco [3], mobile-cloud traffic is projected to constitute 84 percent of the entire mobile traffic in 2017, growing by 14 folds from 2012, which increases at a faster pace than battery capacity. It is a great challenge as well as an opportunity to manage the energy-efficiency of such a huge amount of data transmissions between the cloud and mobile devices.

The problem we consider is the optimized algorithm for making this energy-delay tradeoff. More precisely, the problem can be formulated as a *link selection problem*: given a set of available links (cellular, Wi-Fi access points), determine *whether to use any of the available links to transfer data (and, if so, which), or to defer a transmission in anticipation of a lower energy link becoming available in the future, without increasing delay indefinitely*. Because

it trades off delay for energy, the link selection problem can be naturally formulated using an optimization framework.

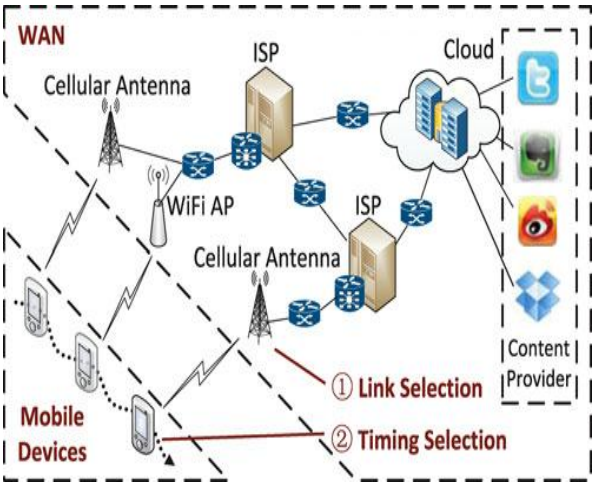

Fig. 1. Using cloud to provide data management for mobile devices.

In this paper, we present a principled approach for designing an online algorithm for this *energy-delay* tradeoff using the Lyapunov optimization framework [8,9]. The link selection problem as an optimization formulation which *minimizes the total energy expenditure* subject to *keeping the average queue length finite*. The Lyapunov optimization framework enables us to design a control algorithm, called SALSA (**S**table and **A**daptive **L**ink **S**election **A**lgorithm) [1] that is guaranteed to achieve *near-optimal* power consumption while keeping the average queue finite. Specifically, in theory, SALSA can achieve power consumption arbitrarily close to the optimal. To our knowledge, prior work has not explored this link selection problem, and our use of the Lyapunov framework for solving this problem is also novel. There are two issues that arise in the practical implementation of SALSA.

First, although control algorithms based on the Lyapunov framework have a single parameter *V*, the theory does not give any guidance on how to set that parameter *V*. We design a simple but effective heuristic for a timevarying *V*, which allows users to tune the energy-delay tradeoff across a broad spectrum.

Second, SALSA requires an estimate of the potentially achievable transmission rate on available link, in order to make its control decision. We devise a hybrid online-offline estimation mechanism that learns link rates with use, but uses an empirically derived mapping between an RSSI reading and the average achieved transfer rate during the learning phase.

The third contribution is an extensive trace-driven evaluation of SALSA using video arrivals from users of our Urban Tomography system and link arrivals obtained from three different locations in the Los Angeles area. The trace-based simulations show that SALSA, which makes its transmission decisions based on three factors, transmission energy, the volume of backlogged data, and the link quality is significantly better than other alternatives that do not incorporate all of these factors in their decisions. Moreover, SALSA's energy-delay tradeoff can be tuned across a wide spectrum using a single parameter a. Finally, SALSA can save between 10 and 40% of the *total energy capacity* of a smartphone battery, relative to a scheme that does not tradeoff increased delay, on many of our video traces. Finally, we validate our trace-based simulations using extensive experiments on a SALSA implementation as part of a video transfer application on the Nokia N95 phones.

II. MOTIVATION AND DESIGN

Saving power of Android enabled devices have become a significant issue with 400,000 such devices being activated daily. Android smartphones and tablets offer several power hungry hardware components and the app developers are exploiting these components at disposal to provide revolutionary user experience. But the battery life has not increased at the same pace to support the power demand. Thus many researchers have been carried out to investigate how to minimize the power consumption in smart phones. Today's smartphones are equipped with high quality graphics and processing power. They have 3G, Edge, Wi-Fi and Bluetooth interfaces for data connectivity. As a result, smartphones have become very popular, and numerous applications are being developed for them. To name a few, surfing the web with browser, calling through VoIP client, checking emails, accessing weather forecast, stock market quotes, and navigating through GPS (Global Positioning System) based maps are some prominent

applications. This increased availability of network applications in smartphones causes increased network traffic, and the volume is increasing rapidly, the growth rate is faster than broadband traffic.

In order to develop energy-efficient networked applications on smartphones, the developers need to know the factors that affect the energy efficiency in wireless data transmission and the joint effects of these factors (network throughput, traffic patterns) on battery life. Existing network management methods have focused on performance of network itself. However there is still need to address the requirements from the perspective of customers and personalized services.

Recent smartphones have multiple wireless interfaces – 3G/EDGE (Enhanced GPRS) and WiFi – that can be used for data transfer. These two radios have widely different characteristics. First, their nominal data rates differ significantly (from hundreds of Kbps for EDGE, to a few Mbps for 3G, to ten or more Mbps for WiFi). The achievable data rates for this ratio depends upon the environment, can vary widely, and are sometimes far less than the nominal values. Second, their energy-efficiency also differs by more than an order of magnitude [6]. While the power consumption on the two kinds of radios can be comparable, the energy usage for transmitting a fixed amount of data can differ an order of magnitude or more because the achievable data rates on these interfaces differ significantly. Finally, the availability characteristics of these two kinds of networks can vary significantly. At least as of this writing, the penetration of some form of cellular availability (EDGE or 3G) is significantly higher than WiFi, on average.

Measurement studies [10,11] show that the energy consumption for transmitting a fixed amount of data is inversely proportional to the available bandwidth. The reason is that the faster a user can download, the less transmission time is needed, and thus less energy is consumed [12]. Note that this calculation does not include the screen-on time if the user is actively waiting for the download process to complete—the screen drains power even faster while a user is waiting. This observation implies that transmitting data in good connectivity could save energy considerably compared to transmitting data during bad connectivity.

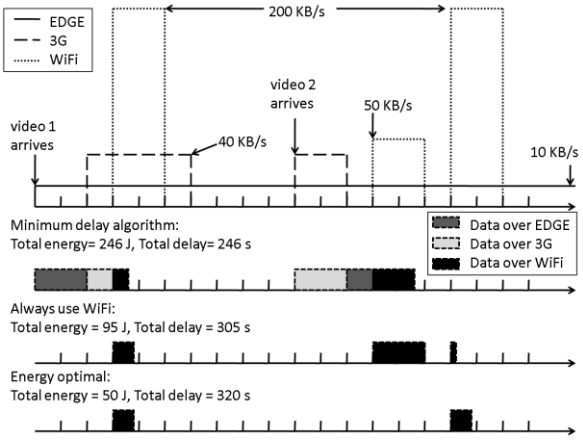

Fig 2 energy-delay trade-off in delay-tolerant, but data-intensive, smartphone applications

Inspired by the considerations mentioned above, energy efficient mobile-cloud communications can be achieved by seizing the "right" timing for data transfers. We therefore envision a transmission protocol for mobile apps that can automatically perceive network conditions and intelligently schedule data transmissions for different apps based on such conditions. Due to the energy saving and potentially downloading data, such a protocol can lead to prolonged battery life and enhanced user experience in mobile devices.

Admittedly, intentionally choosing the time to transmit data may incur delays and affect user experience in real-time or delay-sensitive apps. For example, in SNS apps like Facebook and Twitter, which account for 26 percent of the total app usage time, [4] a user may not immediately check her account when her friends share something, and there is often a time interval between the current and next checking points. Thus, there is an opportunity to prefetch the desired SNS content during good network connectivity. User-generated photos and pictures such as those from Instagram and Viddy constitute a large portion of mobile traffic. Deferring the upload of such content for a short period may not hurt user experience. Furthermore, the synchronization of newly generated mobile data in Dropbox or the downloading of app software updates can also be deferred until good bandwidth is available, since the synced data may not be immediately requested by other Dropbox clients, and most people are not that sensitive to the update time of their apps. In all the cases mentioned

above, SALSA can help flexibly schedule data transfers to conserve energy.

To precisely describe the problem we consider in this paper, let $L[t]$ denote the set of links visible to a smartphone at time t. A link denotes a cellular radio connection (EDGE, 3G or other standard, depending upon the carrier) or a connection to a visible Wi-Fi access point (AP). In general, current smartphone software does not provide applications with the ability to select between different visible cellular radio networks, or control which cell tower to associate with, so we do not assume this capability. However, it is possible, at least on certain smartphone operating systems, to select a Wi-Fi AP for data transfer. $L[t]$ is time-varying: as the user moves, the availability of cellular connectivity will vary, as will the set of visible Wi-Fi APs.

The problem we consider in this paper is the *link selection problem*: if at time t, the smartphone has some data to upload, which link in $L[t]$, *if any*, should it select for the data transfer so as to conserve energy? Our goal in the paper is to design a *link selection algorithm* that solves the link selection problem. One important feature that distinguishes our work from prior work is that the link selection algorithm *can choose to defer the transfer in anticipation of a future lower energy transmission opportunity*. Thus, our link selection algorithm trades off increased delay for reduced energy. Because different applications may have different delay tolerances, our link selection algorithm must provide the ability to control the trade-off.

The link selection problem can be naturally formulated using one of many optimization frameworks. The formulation we choose is based on the following intuition. Suppose that the application data generated on the smartphone is placed in a queue. For delay tolerant applications, it might be acceptable to hold the data in the queue and defer transmission in anticipation of a lower energy link becoming available in the future, but *not indefinitely*. In other words, as the queue becomes longer, it may reach a point where it may no longer be appropriate to trade-off additional delay for energy.

One natural optimization formulation that arises from this intuition is to *minimize the total energy expenditure* subject to *keeping the average queue length finite*. It is this formulation we adopt in the paper, and we introduce a model and associated notation to formally state the optimization objective and constraint. Our model provides a framework for the design and analysis of our online interface selection algorithm. For ease of exposition of the model, we assume that time is slotted; our model and algorithm can easily be generalized to the continuous time case.

III. THE LINK SELECTION ALGORITHM

This algorithm is designed using the Lyapunov optimization framework [8,9], and has the property that it is guaranteed to be stable and can provide near-optimal energy consumption even with varying channelconditions, under some idealized assumptions.

Properties of SALSA

SALSA's control decision [7] is derived using the Lyapunov optimization framework [1]. This framework enables the inclusion of optimization objectives – energy expenditure, fairness, throughput maximization etc. – while designing analgorithm to ensure queue stability using *Lyapunov drift* analysis.Lyapunov drift is a well-known technique for designing algorithms that ensure queue stability. The technique involves defining a nonnegative, scalar function, called a *Lyapunov function*, whose value during timeslot t depends on the queue backlog$U[t]$. The Lyapunov drift is defined as the expected change in the value of the Lyapunovfunction from one timeslot to the next. The Lyapunov optimization framework guarantees that control algorithms that minimizethe Lyapunov drift over time will stabilize the queue(s) and achieve *near-optimal* performance for the chosen optimization objective –for SALSA, power consumption.

We also use another metric, called *dispersion*, to characterizehow far off each algorithm is, on average, from an idealized optimal.For each pair of arrival trace and link trace, we can computethe minimum achievable energy per byte (Em) and the minimumachievable delay per byte (Dm), if each were separately optimized (instead of jointly, as SALSA

does). Specifically, *Em* isthe energy per byte used if all data were transmitted using thehighest rate link in a trace. Similarly, *Dm* is the delay per byte incurred using MINIMUM-DELAY, assuming no transmission failures.For (*Em*,*Dm*) pair, we can also obtain for each algorithm,the Euclidean distance on the normalized $E-D$ plane between theachieved (*E*,*D*) and (*Em*,*Dm*). In general, the latter point may not be achievable by any algorithm that trades-off delay for reduced energy, but it represents a lower bound. For a given algorithm,the average distance of each simulation run from the corresponding "optimal", across all simulation runs, is defined to be the dispersionof the algorithm.

SALSA is also flexible enough to accommodate extensions that may be desirable for smartphone applications. We now discuss two such extensions, but have left their evaluation to future work. In addition to these, SALSA can be extended to accommodate prioritized data transmissions, or bounds on average power consumption.We have omitted a discussion of these for lack of space.

Data Download using SALSA: SALSA can also be extended for link selection for data downloads. Many applications can live with a delay tolerant download capability. Such applications download, in the background, large volumes of data (e.g., videos, images, maps, other databases) from one or more Internet-connected servers inorder to provide context for some computation performed on thephone. A good example is Skyhook's WPS hybrid positioningservice, which prefetched relevant portions of precomputed hotspot location database. To get SALSA to work for such applications, we need to changethe definition of $A[t]$ and $U[t]$. Specifically, we define $A[t]$ as thesize of the request by an application during timeslot *t*, and $U[t]$as the backlog of content that has not been downloaded yet. Inapplying SALSA to the download scenario, we assume that it ispossible to know the size of the content requested by an applicationprior to downloading the content. This is certainly feasible for staticcontent hosted by a server, and for dynamically generated contentfor which the server is able to estimate size.

SALSA for peer-assisted uploads. In a peer-assisted upload, datais opportunistically transferred to a peer smartphone with the expectationof reducing the latency of upload. In general, peer-assistancewill requires the right kinds of incentives for peers to participate.

However, for certain cooperative participatory sensing campaigns, where a group of people with a common objective collectively set out to gather information in an area, peer-assisted uploads are a viable option to increasing the effective availability of network connectivity. For the peer-assisted upload case, we can model the connection to the peer as a link. The important change is that the achievable rate $\mu_l[t]$ of this link takes into account an estimate of the upload delay (the time when the peer expects to meet a usable link). When a Smartphone meets a peer, it queries the peer to get an estimate of $\mu_l[t]$ on the link *l* between them. The peer computes this quantity by estimating the time that it is likely to meet the next AP (say *tm*), and the achievable rate *r* to that AP. It advertises $\mu_l[t]$ as *r/tm* which is an estimate of the effective data rate that would be observed by a transfer handed-off to this peer. Recent work [13] suggests that it might be possible to accurately forecast *r*, and *tm* can be estimated using GPS and trajectory prediction.

IV. CONCLUSIONS

To improve the energy-efficiency of data transmission for mobile devices, some works consider how to select wireless link from multiple available ones. By offloading the overhead of management to the cloud, little burden is placed on the mobile device when saving energy via transmission control. SALSA uses Lyapunov optimization to adaptively make transmission decisions according to current network conditions and queue backlogs.

SALSA is a near-optimal algorithm for performing the energy delay tradeoff in bandwidth-intensive delay-tolerant smart phone applications. Its transmission decisions take several factors into account: data backlog, power cost of the wireless interface, and channelquality. Algorithms which lack even one of these factors in the transmission decisions perform significantly worse. Finally, SALSA solves a real problem: many of the users have collected videos for which the total transmission cost, as wellas the savings obtained by SALSA, are a noticeable fraction of the overall battery capacity. In future work, we hope to get more experiencewith SALSA deployments from our diverse user base.

REFERENCES

[1] M. Ra, J. Paek, A. Sharma, R. Govindan, M. Krieger, and M. Neely, "Energy-delay tradeoffs in Smartphone applications," in Proc. ACM 8th Int. Conf. Mobile Syst., Appl. Services, 2010, pp. 255–270.

[2] Fangming Liu, Peng Shu, and John C.S. Lui, "AppATP: An Energy Conserving Adaptive Mobile-Cloud Transmission Protocol", IEEE Transactions On Computers, Vol. 64, No. 11, Nov 2015, pp. 3051 - 3062

[3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017 [Online]. Available: http://www.cisco.com/, 2012-2017.

[4] Flurry Five-Year Report [Online]. Available: http://blog.flurry.com/?Tag=App+Usage, 2014.

[5] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" IEEE Comput., vol. 43, no. 4, pp. 51–56, Apr. 2010.

[6] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta. "Wireless Wakeups Revisited: Energy Management for VoIP over Wi-Fi Smartphones". In MobiSys'07, 2007.

[7] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. "COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading". In MobiSys'07, 2007.

[8] A. Nicholson, Y. Chawathe, M. Chen, B. Noble, and D. Wetherall, "Improved access point selection," in Proc. ACM 4th Int. Conf. Mobile Syst., Appl. Services, 2006, pp. 233–245.

[9] A. Pathak, Y. Hu, and M. Zhang, "Where is the energy spent
inside my app?: Fine grained energy accounting on smartphones
with Eprof," in Proc. ACM 7th ACM Eur. Conf. Comput. Syst., 2012, pp. 29–42.

[10] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in Proc. 9th ACM SIGCOMMConf. Internet Meas. Conf., 2009, pp. 280–293.

[11] J. Huang, F. Qian, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in Proc. ACM 10th Int. Conf. Mobile Syst., Appl. Services, 2012, pp. 225–238.

[12] Use Wi-Fi Instead of 3G to Save Android Battery Life [Online]. Available: http://goo.gl/HLND7, 2012.

[13] A. J. Nicholson and B. D. Noble. "BreadCrumbs: Forecasting Mobile Connectivity". In MobiCom'08, 2008.