Adapp: An Adaptive Network Selection Framework for Smartphone Applications

Ayon Chakraborty and Samir R. Das
Computer Science Department
Stony Brook University
Stony Brook, NY 11794, USA
{aychakrabort, samir }@cs.stonybrook.edu

ABSTRACT

As smartphones acquire a heterogeneous set of network interfaces provisioned by a variety of providers, smartphone applications have an opportunity to choose amongst multiple services based on the functionality, cost and user-desired quality of experience (QoE). In this work, we propose a framework that predicts the service that suits the application best in terms of QoE while saving energy and dollar costs. We develop a prototype system called 'Adapp,' that trains itself online using user feedbacks and its prediction accuracy improves over use. We demonstrate with the help of rigorous experiments, how different users have varying service preferences for the same application, while the same user can have different preferences across applications. Experimental results validate the adaptive nature of the system. We also analyze Adapp's accuracy in selecting the most appropriate service.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communications.

Keywords

Mobile Internet, quality of experience, mobile apps.

1. INTRODUCTION

A majority of the data traffic in today's smartphones and tablets are driven by a multitude of software applications (apps) running on them. It is instructive to look at how an application's functionality is tightly coupled with its network requirements. Applications that involve real-time editing of shared online documents or interactive multimedia applications can be delay sensitive and/or bandwidth hungry. On the other hand, an application that syncs email, for example, maybe comfortable with moderate or lower quality of services. Some application such as a news ticker or a weather widget may not even require constant network communication as updates could be infrequent. Even performance expected from applications could be very individual or scenario specific. Individuals can trade-off amongst priority, quality of experience (QoE), battery conditions and the dollar cost that the user is willing to pay for the service. With the ever increasing diversity in smartphone usage[1], sticking to a fixed type of service statically does not make much sense from the user's perspective. Across Asia, dual-SIM phones have become popular, while Apple is rumored to be designing a Universal SIM that will allow iPhone users to choose amongst different operators[2]. Cooperation among cellular service providers can bring significant performance improvement [3][4].

With increase in the number and diversity of network service providers and the tiers of services they provide, majority of the users face a dilemma in selecting a service that suits him/her best. Subscribing to a reliable high-speed service may not always be a cost-effective or energy efficient solution. A typical smartphone today can have a number of radio interfaces (viz., 3G, 4G, Wi-Fi and Bluetooth) and the number is expected to grow in near future so that devices with multiple radios are capable of connecting to several networks at the same time [5]. Measurement studies in cellular networks and Wi-Fi show variation in network parameters across different operators, locations and time [6][7]. Operators often deploy their networks in order to optimize different performance metrics, which in turn implies that depending on location and application requirements, the user's choice of best operator could be different [2]. Optimally binding these networks to the phone's interfaces, while giving the user a desired QoE for every application is not a trivial problem.

In this work we address the following problem: Given a smartphone capable of utilizing multiple network services available, how do we allocate a service to an application that saves cost, energy and at the same time satisfies the user. We define the network service by a 3-tuple of network attributes offered by a particular operator, viz., $<$*Average Throughput, Round Trip Time (RTT), Packet Loss*$>$. To this end we develop 'Adapp,' a system that frees users from the dilemma of selecting amongst multiple services for specific applications, while guaranteeing an acceptable QoE. It uses an online learning mechanism based on user feedback. We use Adapp to study several popular smartphone applications, broadly categorized into three types: Multimedia, Interactive and Delay/Disconnection tolerant. Although today's smartphones are equipped with multiple radios connected to multiple networks, the mobile client unfortunately

CellNet'13, June 25, 2013, Taipei, Taiwan

does not have enough support to take simultaneous advantage of all these networks around it. The Adapp system has two components. The first component is the Wi-Fi interface connected via an AP to a controller computer. This AP can mimic different services by controlling the network parameters. The second component is a 'feedback application' that runs on the user's smartphone and can communicate with the AP to switch to a different service on demand. The feedback application is controlled by the user and helps Adapp to continuously learn the service requirements for specific applications. Our contributions include building a complete system prototype abstracting a variety of services and conducting rigorous experiments in a real setting. The main findings of this paper are the following:

1. We observe interesting trends in user-application interaction when multiple services are available. A user's choice of a service varies across applications while users can have different choices for the same application.

2. With the diversity of the user-application interaction[1], design alternatives based on empirical models are not a good option. We suggest that such a framework should be more personalized rather than generic.

The rest of the paper is organized as follows. Section 2 introduces the modeling framework and methodologies used to build and analyze the system. Experimental setup and data collection procedures are described in Section 3. We present the results in Section 4 along with our observations. Section 5 concludes the paper.

2. METHODOLOGY

This section forms a modeling framework on which our experiments are based. First, we describe our basis on choosing the service levels. Second, theoretical formulations for modeling user benefit are presented. Third, we describe an algorithm for the optimal selection of the service levels.

2.1 Choosing Service Levels

We use actual service levels offered by real service providers. We took real performance measurements from different 3G/4G and Wi-Fi networks from the users' smartphones accessing those services. Each such measurement can potentially represent a service level. This allows us to mimic a real world operator in the experiments and separately examine the impact of individual network parameters.

Metrics: The network parameters that we accounted for are average throughput (both uplink and downlink), round trip time (RTT) and packet loss rate. These three parameters are used to represent a service level. Details of these measurements and the exact choice of service levels are described in Section 3.1.

2.2 Modeling User Benefit

Adapp aims to improve the user benefit by optimal selection of a service level. We use the general concept of utility (U) [8] of a flow that can be expressed as a function of network parameters (e.g., throughput). Our model also utilizes the user's preference for a particular service level over other available services as input to the utility function. It is evident that the user's preference is highly dependent on the service's QoS parameters. Also each service has a cost (C) associated with it. The latter can be thought of a combination of energy cost (of the radio interface) and/or dollar cost required to access the service. The difference of U and C models the user benefit B that we want to maximize. We express the user benefit B as,

$$B = U - C_{dollar} - C_{energy}. \tag{1}$$

2.2.1 Modeling Utiltiy

We assume that the user is aware of the dollar cost (which is often true), and also a better QoS generally incurs a greater cost. Given two service levels that satisfy the user's requirements, he/she prefers the one with lower dollar cost. Our utility model incorporates this idea.

Hypothesis 1: *The more time a user uses a service level, the more is his preference for that service level and higher is its utility.*
We will use the above hypothesis in deriving our utility function. For each application A_i, and for each service level S_j, Adapp keeps track of the following measurements.

$\boldsymbol{T_{ij}}$ **:** The total amount of time application A_i utilized service S_j.

$\boldsymbol{F_{ij}}$ **:** The fraction of total time application A_i utilized service S_j. F_{ij} can be expressed as, $F_{ij} = \frac{T_{ij}}{\sum_j T_{ij}}$. It can be thought of as a measure of the user's preference for S_j over other available services or the utility of service S_j for the application A_i (U_{ij}):

$$U_{ij} = F_{ij} = \frac{T_{ij}}{\sum_j T_{ij}}. \tag{2}$$

We demonstrate the above notations in the following example, for a single application X, and a pair of service levels A and B.

A		B			A	B			A
1	2	3	4	5	6	7	8	9	10

Figure 1: The above figure shows two service levels A and B used by an application X, with a timeline (10 time slots). $T_{XA} = 4$, $U_{XA} = F_{XA} = 0.4$, $T_{XB} = 6$, $U_{XB} = F_{XB} = 0.6$.

A costlier service suffers a lower U_{ij} value due the fact that although it satisfies the user's requirements but he can achieve the same level of satisfaction with a lower costing service. Thus, comparatively, he/she spends lesser amount of time in a costlier service. Equation (2) implicitly takes care of the dollar cost.

2.2.2 Modeling Cost

C_{dollar} already taken care of by the utility model. We evaluate C_{energy}, the energy cost parameter associated with a service.

Energy Cost: Some recent works [9][10] perform a detailed measurement study and modeling of energy consumptions for 3G, 4G, GSM and Wi-Fi networks. We adapt to the formulation of [10] for modeling energy costs of these cellular data networks. For simultaneous uplink and downlink transfers, considering separate receivers and transmitters, power consumed (in mW) is given by:

$$P = \alpha_u t_u + \alpha_d t_d + \beta \tag{3}$$

In (3), t_u and t_d denotes uplink and downlink throughputs. β is the base power when throughput is 0. Individual power consumptions for receiving and transmitting can be given by, $P_u = \alpha_u t_u + \beta$ and $P_d = \alpha_d t_d + \beta$ respectively. The best-fit values as shown by [10] are given in Table 1.

Technology	α_u(mW/Mbps)	α_d(mW/Mbps)	β
4G	438.39	51.97	1288.04
3G	868.98	122.12	817.88
Wi-Fi	283.17	137.01	132.86

Table 1: Empirical models for energy consumption in mobile devices for 4G, 3G and Wi-Fi technologies.

As seen from the figures in Table 1 and considering average throughput rates of 3G, 4G and Wi-Fi connections (Table 2), we can infer 4G and 3G technologies consume much more energy than Wi-Fi, while 4G is more power hungry than 3G. We calculate Energy Cost, C^i_{energy} for the service S_i as

$$C^i_{energy} = \frac{P^i_{energy}}{P^{max}_{energy}}. \tag{4}$$

P^i_{energy} denotes the power consumed when using service S_i while P^{max}_{energy} denotes the power that would have been consumed while using the highest service level (in terms of average throughput) in the same technology (say, Wi-Fi, 3G or 4G).

2.3 Service Level Selection

Assume a smartphone running k applications $\{A_1, A_2, \cdots A_k\}$ and has access to N different service levels $\{S_1, S_2, \cdots S_N\}$, where A_i represents the application id and S_i represents the service id. We also assume that the smartphone user operates one of these k applications at a particular time. Whenever a user runs an application for the first time, Adapp allocates it the service having the lowest service score (of course the cheapest one). The user switches to the most suitable network by using the feedback application. Adapp starts storing the information as mentioned in the previous subsection and uses it in the service selection algorithm. It chooses the service for which the user benefit (1) is maximized. Figure 2 shows a generic idea of our Benefit model. The utility value peaks at a certain service level indicating the threshold requirement of the user. Better service levels incur greater dollar cost and hence lower utility values.

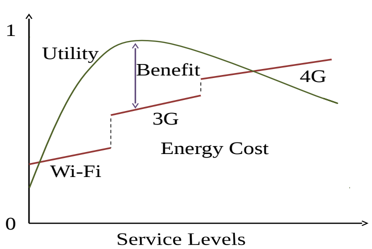


Figure 2: Adapp maximizes Benefit, which is the difference between the Utility and Energy Cost.

Everytime an application is run, Adapp keeps track of the utility parameters (mentioned in 2.2) and recalculates the new values of B_{ij}. When the application is rerun for the next time, the new B_{ij} values are taken into account for the service selection algorithm. Thus the service selection algorithm forms an online prediction model that gets better on use.

Algorithm 1 Optimal Service Level Selection

Input: F_{ij}, C^i_{energy}
Output: $N = \{N_1, N_2, \cdots N_k\}$ /* vector of service allocations */

for all applications A_i
  $max\_benifit_i \leftarrow 0$
  for all available service levels S_j
    $B_{ij} \leftarrow F_{ij} - C^i_{energy}$
    if $max\_benifit_i > B_{ij}$
    $max\_benifit_i \leftarrow B_{ij}$
    $N_i \leftarrow j$ /*allocate j^{th} service for i^{th} application*/
    endif
  end for
end for
return N

3. EXPERIMENTS

In this section we describe experiments we performed to emulate the service levels and details of the experiment testbed. We also describe the data collection procedure including user experiments we performed.

3.1 Emulating Service Levels

A commodity smartphone today uses a single service provider with a specific level of service (for example, a Verizon 4G or a Sprint 3G connection). As discussed in Section 1, our aim in this paper is to make optimal use of the diversity of services that may be available at particular location. Thus, we took into account several cellular data services and Wi-Fi networks locally available in the Stony Brook University campus imagining that a smartphone can potentially receive services from all these networks.

Measurement Experiment: For the cellular data service providers we have considered the 4 major carriers in the US, viz., AT&T, Sprint, Verizon and T-Mobile. Several volunteers using those services in their phones were appointed for measurement studies. The experimental setup for the measurement consists of a wired server residing at the Stony Brook University and several mobile clients. We created a mobile application that the volunteer uses to download a 10MB file from the server in his/her phone and uploads it back to the server, repeating this at different times of the day. In all experiments we collect traces from the server using *tcpdump* and *tshark* utilities. We verified that the CPU utilization caused by trace collection is lesser than 5%. Table 2 shows the information collected from the different networks about their service levels. We ranked the service levels (from 1 through 10, a higher rank means better/costly service) based on their service scores (defined below). Although for a given service provider, we observed fluctuations in performance but they were not high enough to change its rank.

Service Score: We define Service Score of the i^{th} service S_i as,

$$\sigma_i = \frac{1}{4}\left(\frac{T^u_i}{T^u_{max}} + \frac{T^d_i}{T^d_{max}} + \frac{D_{min}}{D_i} + \frac{L_{min}}{L_i}\right) \tag{5}$$

In (5), T^u_i, T^d_i, D_i and L_i stand for average throughput (uplink and downlink respectively), RTT and loss for S_i. T^u_{max} and T^d_{max} stand for maximum uplink and downlink throughput, D_{min} and L_{min} stand for minimum RTT and packet

Rank	**Carrier**	**Technology**	**Throughput** (Mbps)		**RTT** (ms)	**Loss Rate** (%)	**Service Score**
			Uplink	*Downlink*			
1	SBU Wi-Fi	802.11	0.11	0.13	193	0.31	0.234
2	SBU Wi-Fi	802.11	0.17	0.24	167	0.28	0.272
3	Sprint	3G	0.46	0.39	229	0.26	0.282
4	AT&T	3G	0.43	1.26	182	0.23	0.358
5	T-Mobile	3G	0.72	1.58	201	0.27	0.362
6	Verizon	3G	0.49	0.52	147	0.19	0.389
7	Sprint	4G	0.79	2.28	147	0.25	0.443
8	T-Mobile	4G	0.61	2.96	129	0.23	0.490
9	Verizon	4G	3.08	3.13	124	0.18	0.749
10	AT&T	4G	2.47	5.67	76	0.15	0.951

Table 2: The above table provides idea about the diversity of services that can be available at a given location. The measurements were taken in our university campus. It is assumed that the cost increases from a lower service level to a higher one. These parameters were used to emulate the above mentioned networks.

loss across all the service levels. Theoretically, σ varies between 0 and 1, where a higher σ indicates a better service. Service Scores for the emulated service levels has been presented in Table 2, on which their rankings are based. These ranking indicates an implicit ordering of the service levels in terms of QoS.

User Mobility: Although our measurements did not span multiple geographical regions, the work can be easily extended to dynamic scenarios where the user moves frequently. In mobile situations Adapp updates itself with the new list of services that are currently available. Statistics for the services that were previously available but currently unavailable are logged for future use.

3.2 Experimental Setup

This subsection describes the testbed for our experimental setup. It consists of a controller computer having both wired and wireless interfaces, and several mobile clients running the feedback application. The wired interface of the controller computer (Dell Optiplex Server, running Ubuntu Linux 11.04) is connected to an Intel Gigabit Ethernet LAN belonging to the university network. We have inserted an Atheros-C9 wireless card in the PCI slot of this computer and configured it as an access-point, which is then bridged to the wired interface via IP masquerading.

The controller runs Netem [12] – a Linux network emulator module that can control network parameters (delay, loss, duplication, corruption etc.) for testing and analyzing protocols by emulating network properties. Although Netem does not provide direct hooks to control throughput, packet rate can be controlled by using other queuing disciplines [13]. We configured Netem to emulate the 10 service levels as discussed in the previous subsection. Netem has the capability to emulate different network parameters for different IP addresses that makes it possible to run multiple mobile clients with different service allocations. The emulated performance is verified to match closely with the target performance indicated in Table 2.

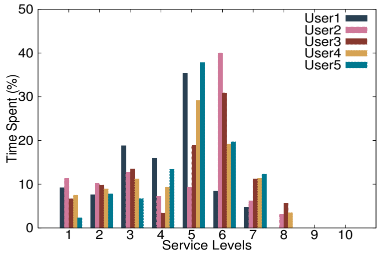
3.3 The Feedback Application

Although an initial ranking based on network performance is done, the specific choice of a particular service depends on the user. It is not necessarily the best service that is required but the most optimal one which satisfies the user's requirements. The feedback application helps Adapp to learn about the user's preference by allowing him to switch in between service levels. It is developed for the Android platform and has three control buttons: Boost, Degrade and Reset. The 'Boost' button switches to the service level with the next higher rank, while 'Degrade' switches it to the next lower rank. 'Reset' selects the lowest ranked service (i.e., 1). It gives a lot of leverage to the end user and makes Adapp adaptive. The feedback application sends a specially coded UDP packet to convey the user's action to the controller computer. This can be thought of as a separate control channel between the mobile client and the controller. The controller has a service-rank mapping and it starts emulating the requested service. When the user is not using any application assuming email sync, news update widget etc are still on, Adapp reverts back to the default service level.

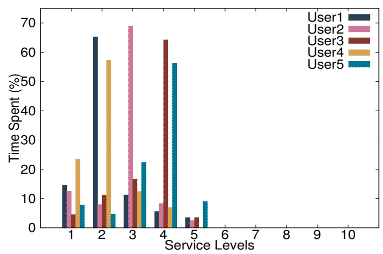
3.4 User Experiments

Adapp was tested on five different subjects for two weeks (12 days except Sundays). We chose five because beyond this limit we were not able to simultaneously provide the highest service level to everybody due to bandwidth restrictions. The subjects were selected such that they spend most of their working hours in the vicinity of our lab (e.g., neighboring labs, graduate offices) where Adapp's access-point is installed. Their smartphones were running Android v4.1.1 (Jelly Bean). The subjects were instructed to connect specifically to this access-point (password protected to avoid unwanted traffic) during this period, and were assigned fixed ip addresses. The feedback application was pre-installed in all the mobile clients. According to a small pilot survey conducted before the experimentation, we decided upon a set of applications that were popular and more commonly used by the subjects. We chose some Multimedia applications (Youtube, Skype), Interactive applications (Google Maps) and other popular applications (Facebook, Gmail). Traffic from other applications were ignored in the study.

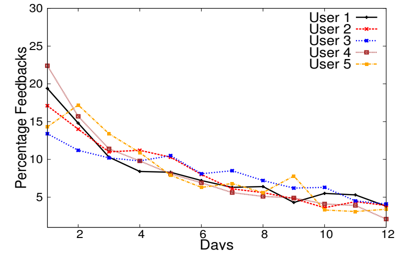
Whenever the subject starts an application it is allocated the pre-computed service level (Section 2.3). This is the default service (the rank one service i is used for the very first invocation). If the user is unhappy with the current service and willing to opt for a better one he presses the 'boost' button in the feedback application. The user is aware of the fact that pressing the 'boost' button will allocate him a service with a greater cost than the current one. Similarly, for a conservative user who can make a tradeoff in performance for a lower cost can 'degrade' to a lower service. The controller computer does the required bookkeeping for the utility parameters. It also keeps track of the uplink and downlink throughputs (t_u and t_d) in order to measure energy costs.



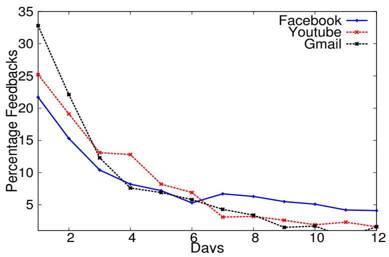
(a) Service Choice - You Tube



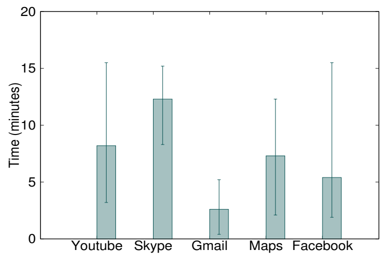
(b) Service Choice - Facebook



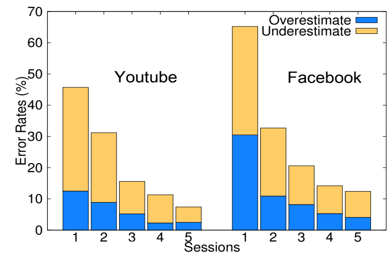
(c) User Adaptation



(d) Application Adaptation



(e) Session Lengths



(f) Error Estimates

Figure 3: Figures (a)(b) show the percentage of time spent by users in different service levels for applications Youtube and Facebook. Figure (c) shows the user adaptation during the experimentation period while Figure (d) shows the adaptation for a particular user across different applications. Figure (e) presents the average session lengths (along with max-min values) for a user across different applications. The error rates are presented in Figure (f) for Youtube and Facebook.

Thus whenever an application is used, the preferred service level for that application gets updated. The more an application is used, the more it adapts to maximize the user's benefit.

4. RESULTS

In this section we present experimental results that validate the usefulness of Adapp. The user experiments were done in a natural setting where the subjects to do their daily business in their smartphones. Due to space restrictions it is not possible to show results for every application we studied but we present some representative results supplemented by a broad discussion.

4.1 Choice of Service

Figures 3a and 3b show the distribution of usage time of the service levels for two applications (Youtube and Facebook). It is evident from the distribution that users prefer to use a particular service more compared to the others, while using a certain application. Perhaps what is more interesting is that, not only does the distribution vary with different users but also it varies for the same user across applications. Hence propositions to model QoE for different applications through empirical formulations do not seem to be reasonable. Our findings show, that these formulations should be rather personalized than generalized. However, the applications – depending on its category – may show certain general trends, but this is not the case always. For example, streaming video applications (like Youtube), shows an approximate pattern - users show more preference for services 5 and 6 (in figure 3a). On the other hand, we cannot derive any commonality among the users using the Facebook application.

4.2 User Adaptation

We show how Adapp gradually adapts to the user's behavior for the slot of time we performed our experiments. As stated earlier, the user uses the feedback application to switch to a different service when he is not contented with the service level he is being offered. More is the feedback, more Adapp learns about the user's *comfort zone* and more accurate is the service allocation. One indication of this learning is the user issuing less feedbacks from the mobile client. In figure 3c, we plotted the distribution of feedbacks for the users from Day 1 through 12. As we see in the plot (3c), out of the total number of user feedbacks, the majority came during the initial period and shows an overall decreasing trend. This decreasing trend is an indicator of Adapp's adaptability. Unless the user completely changes his behavior or relocates to a new place with completely different services, the adaptation performs well. In figure 3d we analyze the distribution of feedbacks for a particular user for popular applications like Facebook, Youtube and Gmail. Here also we see how the application type plays a specific role in the adaption process. This trend also varies across users.

4.3 Estimation Errors

Adapp is an online service level prediction system. As with the case of any prediction system, an important aspect is to estimate its accuracy. Adapp's accuracy lies in the fact that how optimally it can judge the user's desired service level. Allocating a service level at a higher rank than required causes cost penalty, which we call overestimation error. Similarly choosing a poorer service level than required may hamper the user's experience, which we call underestimation error. For each session the user starts us-

ing an application, Adapp allocates a service level for it. A session is the time the user starts using the application before he/she leaves the application or switches to another application. Figure 3e shows a user's mean session lengths (along with maximum and minimum values) across different applications.
Figure 3f shows the error rates for five such sessions for the user using Facebook and Youtube applications. The Youtube sessions roughly lasted for 8 minutes where as the Facebook sessions were around 5 minutes each. The sessions were selected such that they are equally distributed over the experimentation period; say for Facebook, sessions 1 and 5 were selected from days 1 and 12, and sessions 2,3 and 4 were selected from days 4,7 and 10 respectively. A similar strategy was taken in choosing the Youtube sessions. The fraction of time in a session, the user spends in other service levels (changing it through the feedback application) is called the *estimation error*. The error rates show a receding trend over time. An interesting observation is that the overestimation errors were less compared to the underestimation errors. The Service Selection Algorithm (Section 2.3) maximizes the user benefit; hence it always tries to choose the lowest ranking service that satisfies the user saving energy and cost.

4.4 Observations

It is not the user or the application alone, but the user-application pair plays a important role in determining the optimal service requirement. Each application has its own unique nature regarding how it gets adapted to the user. Application with highly variable session lengths (eg., Facebook) gets adapted slower in comparison to others (In figure 3f, error rates of Facebook are more than Youtube). With the emergence of multiple cellular data network services in smartphones, and the array of applications available, it is vital to decide the service allocations among applications. Such a system saves energy (vital for smartphones) and dollar costs while guaranteeing a good user experience.

The limitation of a system like Adapp is that the adaptation performance decreases in highly mobile situations. A mobile user travelling over a larger geographic region may frequently see newer services and it may so happen that the user shifts to a new location even before the system learns about the user's preferences. Our future work will concentrate in dealing with these highly mobile scenarios. Another future work is involving sophisticated learning algorithms for Service Selection and investigate how it improves Adapp's performance.

5. CONCLUSION

We developed a prototype system, Adapp, that optimally selects a network service level among a set of available services (with different QoS) adapting to the user's preference (maximum benefit). We tested our prototype for different users across several popular applications and observed interesting trends in the user-application interaction. The results demonstrate that Adapp indeed adapts to the right service level with receding error rates over time. We promote a personalized and adaptive framework in building such a system rather than a generalized one based on empirical models. A framework like Adapp gives users the complete flexibility in 'making use of all the networks' around them [5] and on the same time saves energy and cost. Users vary as do applications and Adapp draws a connecting link between them.

Acknowledgements

This work was partially supported by NSF grants CNS-0831791 and CNS-1117719. The authors also acknowledge the help extended by the volunteer subjects in the user experiments.

6. REFERENCES

[1] H. Falaki et. al. Diversity in smartphone usage. In *Proc. MobiSys*, 2010.

[2] S. Deb, K. Nagaraj, and V. Srinivasan. MOTA: Engineering an operator agnostic mobile service. In *Proc. Mobicom*, 2011.

[3] C. Singh, S. Sarkar, A. Aram, and A. Kumar. Cooperative profit sharing in coalition-based resource allocation in wireless networks. In *Networking, IEEE/ACM Transactions on,* 2011.

[4] P. Lin, J. Jia, Q. Zhang, and M. Hamdi. Cooperation among wireless service providers: opportunity, challenge, and solution. In *Wireless Communications,* IEEE, vol. 17, no. 4, August 2010.

[5] Kok-Kiong Yap et. al. Making Use of All the Networks Around Us: A Case Study in Android. In *Proc.In CellNet*, 2012.

[6] Chen, Yung-Chih, et al. Measuring Cellular Networks: Characterizing 3G, 4G, and Path Diversity. *UMass Amherst Technical Report: UM-CS-2012-022.*

[7] Wee Lum Tan. An Empirical Study on the Capacity and Performance of 3G Networks. In *IEEE Transactions on Mobile Computing* , vol. 7, no. 6, June 2008.

[8] Scott Shenker. Fundamental Design Issues for the Future Internet. In *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, September 1995.

[9] Balasubramanian, N., Balasubramanian, A., and Venkataramani, A. Energy consumption in mobile phones: A measurement study and implications for network applications. In *IMC*, 2009.

[10] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A Close Examination of Performance and Power Characteristics of 4G LTE Networks. In *Proc. Mobisys*, 2012.

[11] Costas Courcoubetis and Richard Weber, Pricing Communication Networks, Economics Technology and Modeling. Wiley-Interscience series in Systems and Optimization.

[12] Hemminger, S. Network Emulation with NetEm. In Linux Conf Au, 2005.

[13] Bandwidth Throttling with NetEM Network Emulation http://www.linuxforu.com/2012/06/bandwidth-throttling-netem-network-emulation/.