Context-aware Handoff on Smartphones

Qiang Li
Institute of Information Engineering
University of Chinese Academy of Sciences
Beijing, China
Email: liqiang@is.iscas.ac.cn

Qi Han
Department of EECS
Colorado School of Mines
Golden, CO USA 80401
Email: qhan@mines.edu

Limin Sun
Institute of Information Engineering
University of Chinese Academy of Sciences
Beijing, China
Email: sunlimin@iie.ac.cn

***Abstract*—Nowadays smartphone users often enjoy the availability of multi-networks by switching between the networks for better network performance, energy efficiency of smartphones, and more data offloading to less expensive networks. However, network switching inevitably brings about network disruptions leading to user experience degradation. In this paper, we propose an application context model that is used in conjunction with a heuristic network selection mechanism, which selects a network using three metrics (i.e., network performance, energy consumption, and cost). A Bayes classifier is used to provide a probability for the network selection given applications running during network disruptions. We construct the classifier via crowd-sourced data by considering smartphone users profile and the operating environments. We implement a prototype context-aware handoff on the Android platform and conducted an experiment in a real-world scenario through one case study, switching between cellular and WiFi networks. The evaluation results suggest that context-aware handoff achieves 25% energy cost, nearly one-third data offloading, and more than twice throughput with only one third of the network switchings.**

I. Introduction

Current smartphones have multiple wireless network interfaces that have different energy cost, throughput, and coverage area. For instance, wireless wide area networks such as GPRS/WCDMA provide nationwide coverage but with scarce cellular bandwidth and high spectrum cost; in contrast, wireless local area networks such as WiFi offer limited coverage but with relatively higher bandwidth and lower cost. Typically, lots of research investigates how to properly exploit multi-networks to improve network performance [1], energy efficiency [2] of smartphones, or offload data [3] to less expensive networks. Despite these benefits, network switching leads to undesirable disruptions of ongoing network flows from the perspective of end-users. If not handled properly, this network disruption may be irritating and frustrating and users may no longer want to reuse the service of exploiting multi-networks because of the poor experience.

Nowadays, a smartphone's handoff among multi-networks is different from that on early portable devices such as cellphones, PDAs, and laptops. The biggest difference is that applications on a smartphone is like a gateway for people to access networks. When a smartphone is pre-installed with marketplace portals such as AppStore on iOS, Google Play on Android, and MarketPlace on Windows Mobile, a user can easily discover and download applications with different network interfaces. Different applications (such as pervasive web browsing, streaming video, and VoIP applications) react differently to a network interruption. For instance, when WiFi switches to the cellular network, web browsers on the smartphone has the caching scheme for the network disconnection so that users will not notice network interruptions. In contrast, online multimedia streaming restarts the service as network IP changes, so users easily notice this disruption. In other words, smartphone applications play the principle and direct role on the choice of network interface, while other factors such as bandwidth, delay, and jitter serve merely as initial and indirect role. We believe the choice of which network to use directly depends on applications. Being application-aware will improve user experience during network switching while fully exploiting multi-networks.

However, constructing an application-aware model for network switching is faced with interesting challenges. First, the number of applications is huge. For example, Android has 150K apps and 350K daily activations [4], and more than 350K apps are available on the AppStore [5], up to 2012. Further, applications determining networks selection have different built-in factors such as application protocols (e.g. HTTP or FTP), network status indicators (e.g. bandwidth or jitter), certain schemes against network fluctuations (e.g. caching or re-connect), and interactive mode with users. It is impossible and unnecessary to classify and enumerate the criteria of all applications when faced with network fluctuations caused by network switching operations. Second, application usage of smartphone users is uncertain. In addition, there are always new and unforeseen applications on the market, requiring the application context model to evolve over time to ensure accuracy.

Consider the following scenario, Alice's smartphone has built a context model for 15 applications installed for her daily usage, and Bob's smartphone has 19 applications and does not have a context model. Bob and Alice have the same 10 applications, so by sharing the application data from Alice, we can build Bob's context model without a training process that needs to collect the history of Bob's application usage. Inspired by these observations, we propose to construct an application context model using crowd-sourced application data. This is an evolved form of the crowdsourcing concept [6], soliciting contributions from a large group of people. However, the use of crowd-sourced application data must be judiciously done. They must only selectively be used during sharing so

978-0-7695-5104-3/13 $26.00 © 2013 IEEE
DOI 10.1109/MASS.2013.32

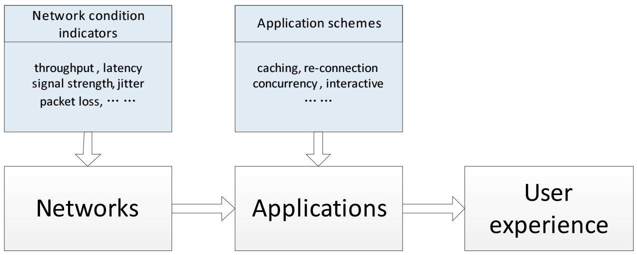


Fig. 1: Factors affecting user experience on smartphones

the model is suitable for individual usage. We propose to use similarity features to partition people into different groups in crowds. In the same group, people share the application data to train and adapt an application context model. We further combine the application context model and a handoff policy to exploit benefits of multi-networks and guarantee user experience.

In brief, our work makes the following contributions.

1) We design a handoff policy consisting of two parts: a heuristic network selection scheme and a context model. The heuristic network selection scheme only takes three metrics of the network selection criteria: network performance, energy consumption, and cost. A context model is a Bayes classifier that provides a probability for the network selection given applications running during network disruptions.
2) We collected training data of the context model via crowd-sourced data by considering smartphone usage similarity and user geo-label similarity in crowds. A hierarchical clustering scheme is used to divide people into different groups. In the same cluster, users have a higher probability to share applications data, and we use the shared data to construct the Bayes classifier.
3) We validate context-aware handoff through one commonly seen case, switching between cellular and WiFi networks. We have implemented context-aware handoff on the Android platform and evaluated it in a real-world scenario. The context-aware handoff policy achieves only 25% energy cost, one-third data offloading and more than twice throughput compared with switching networks on state-of-the-art smartphones, meanwhile preventing user experience degradation caused by network fluctuations.

II. An Exploratory Study of Smartphone Handoffs

In this section, we explore two aspects of smartphone applications from the perspective of user experience: *(1) what factors affect user experience during multi-networks switching? (2) How to construct an application context model that fits more people?*

A. User Experience Degradation

Without the support of additional infrastructures or modification of network protocols, network switching in a brute-force manner inevitably causes user experience degradation. Previous work [7]–[9] adopts network layer indicators to determine network selection opportunity. However, this is no longer sufficient. The reason is that applications now play a dominant role in the daily usage of smartphones and applications act as gateways between users and networks, leading to a big gap between network layer indicators and user experience. As shown in Figure 1, applications performance directly affects user experience when network fluctuations occur during network switching, and network conditions indirectly affect user experience via affecting applications performance. Therefore, directly adopting network condition indicators (e.g., throughput, latency, signal strength, jitter, and packet loss rate) to measure the degree of user experience is not accurate.

To understand the impact of applications performance and network conditions on user experience, we conducted a study that records network conditions and changes of applications during the switching between WiFi and Cellular. We characterize user experience degradation by considering user response time and significant interferences users perceive. As shown in Figure 2, packet loss rate and frame retransmission rate are caused by the network fluctuations during network switching, different applications react differently leading to different response time. We observe that applications have different impact on network performance during switching (Figure 2a). For instance, an application for file updating needs more time for downloading than an email application even with the same packet loss rate caused by network disruptions. We also

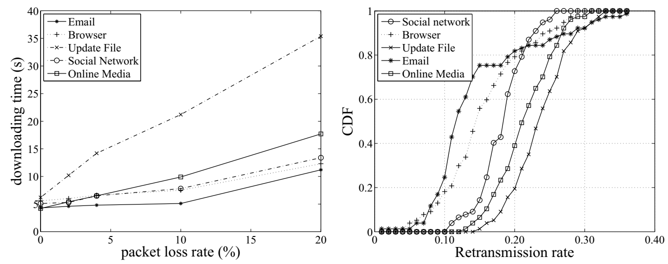


Fig. 2: User experience degradation during network switching: (a) download time of five applications during network fluctuations; (b) CDF of retransmission rate of five applications during network fluctuations.

Application Category	Mechanisms	Interaction Frequency	Annoying
Email	reconnect concurrency	low	no
Browser	reconnect, session caching, concurrency	medium	N/A
Update File	concurrency	low	yes
Social Network	reconnect, session concurrency	high	yes
Online Media	reconnect, buffer concurrency, session	high	yes

TABLE I: Application mechanisms for improving user experience

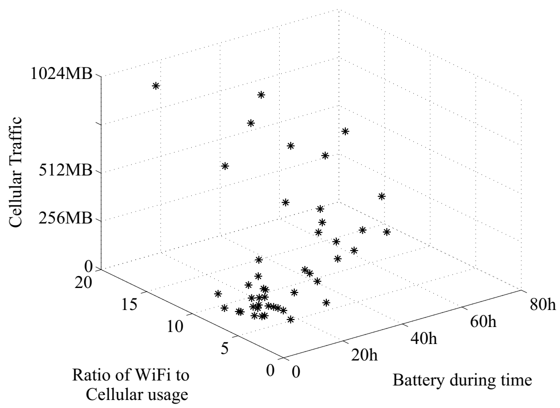


Fig. 3: Smartphone usage

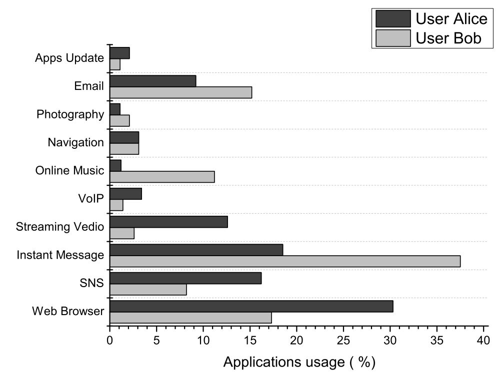


Fig. 4: Applications usage of two people

observe that the CDF of different applications on data frame retransmission rate during network interruptions is distributed (Figure 2b). The reason is that different applications adopt different schemes such as caching, support of session and concurrency to deal with network interruptions as shown in Table I. For example, concurrent multi-network flows in a browser application will reduce web page download time to improve user experience but would lead to more packet losses when network switching occurs. These application schemes would bring different user experience, and only network layer indicators cannot fully reflect user experience. Furthermore, nowadays most applications have user Interface for interacting with users. The interactive mode as shown in the third column in Table I has a great impact on user experience. For example, a web browser interacts with a user more frequently than an email application at the interactive mode.

In summary, we consider both applications and network conditions as network selection criteria in order to guarantee user experience while enjoying the benefits of multi-networks.

B. Application Context Model

If we had the evaluations of all applications for network switching operations as shown in Table I, a context model could have been directly constructed. Most previous work [10] on handoff assumes that applications assessments are done *a-priori*. However, there is a large number of applications on the market, so the assumption does not hold. It is impossible and unnecessary to classify all applications on the market, because the average number of applications used daily by an individual is only a few dozen, much less than the total number of applications available.

Previous work [11] claims that the choice of which applications to use depends on smartphone users profile (e.g., preferences, needs, location) and the environment where she is operating (e.g., time, energy cost, network traffic). Therefore, we conduct controlled experiments in our lab: we collected smartphone usage traces of 45 participators from our lab. The traces include four aspects: battery duration, cellular network traffic, and the ratio of WiFi to cellular usage, and application usage preferences. Each participant carries a mobile phone, where a logging service we implemented runs at the background for data recording. We explore whether there are similar application usage preferences among people that may be used to co-train a application context model.

First, we analyze the data from three dimensions: battery duration time, cellular traffic, and the ratio of WiFi to cellular usage, as shown in Figure 3. The reasons that we choose these three measurements are that battery usage reflects energy usage of a smartphone, cellular traffic measures the condition of data offloading, and the ratio of WiFi to cellular usage reflects the changes in wireless conditions of users. We observe that disparity and similarity co-exist among people. In other words, some people have similar smartphone usage, and some do not. We use Euclidean distance between two participants to indicate the similarity of smartphone usage. We select two users from 45 participants and compared usage preferences of the same application as shown in Figure 4. It shows that users Alice and Bob have more than 10 identical application categories although they were randomly selected. They both use web browser, social networks and instant messaging applications a lot. However, they have different application usage. Comparing with Alice, Bob uses more instant text messaging applications, online media, and email. Although there are a lot of applications on the market and their subjective assessments are unknown, people have a certain degree of similarity and disparity for applications usage.

This motivates us to believe that users with similar application preferences may share training data for network choices and that shared data can be used to build a context model.

III. Context-Aware Handoff Design

In this section, we provide an overview of the context-aware handoff architecture as shown in Figure 5. The architecture is implemented on smartphones, only within the software level, without modifying network protocols and infrastructure. It consists of three principal components: a monitoring daemon, a network selection scheme, and an application context model. The monitoring daemon collects a variety of context information about network conditions, smartphone power data, traffic volume, and application data. The context information will be used as input for the network selection scheme and the context model. The network selection scheme uses three metrics to guide network selection: network performance, energy consumption of network access, and cost of network access, while also considering user preferences. A context model is a naive Bayes classifier that classifies network types with probability.

A. Monitoring Daemon

There are a number of network selection criteria that may be used to determine the best switching time and the most appropriate network, from network to application requirements, from user preferences to environmental conditions. Some may be assumed static and fixed data, such as user/mobile phone profiles (requirements, maximum price allowed, and device hardware/software capabilities). Others are dynamic, such as network bandwidth, delay, QoS levels, and costs. The monitoring daemon is responsible for monitoring and collecting dynamic context information for context-aware handoff. This context information should be widely acceptable, easy to obtain, and expressive. Our monitoring daemon collects four types of context information: (i) network conditions: RSSI or SNR variations and available network bandwidth; (ii) power data: the state of battery and its remaining capacity, application battery consumption; (iii) traffic volume: the amount of transmitted data for each application and each network type; (iv) application data: application description, identified tag, and category of running applications. The network selection scheme component uses these dynamic context data as input to trigger handoff, and the context model uses these data as features to classify network types with probability.

B. Network Selection Scheme

Three metrics are used for network selection: network performance, energy consumption, and cost of accessing a network. These metrics are what most users typically care about most. Dynamic context data from the monitoring daemon may be converted into these metrics. (a) For the network performance metric, available network bandwidth is used. For example, in WLAN, RSSI values and throughput are used to derive available bandwidth of APs; in cellular networks, RSSI values are mapped to pre-defined values of performance. (b) For the energy consumption metric, we compute total energy consumption as the summation of energy consumption during transmission and being idle. Average energy consumption for each network type is empirical estimated and measured in advance. (c) For the cost metric, we calculate the volume of downloaded data and use the billing scheme of each network type that is statically defined. After calculating all the three metrics, we normalize the network selection criteria via a

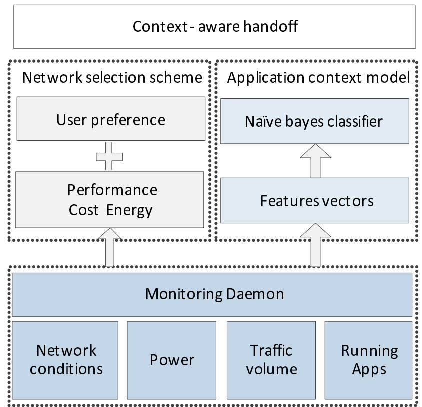


Fig. 5: Context-aware handoff architecture of smartphones

utility function as follows:

$$u(x) = \begin{cases} \frac{1}{1+(\frac{X_0-\theta}{\theta-x})^{\zeta}}, & x \geq X_0 \\ 0, & otherwise \end{cases} \quad (1)$$

where θ is an empirical point of the utility (e.g., $u(\theta) = 0.5$), X_0 is the limited lower bound value, and ζ is the tunable steepness parameters indicating user sensitivity.

We use user feedback as the user context input to weigh the three network selection criteria:

$$\overrightarrow{W} = \{w_i | w_i \in [0,1], \sum_{i=1}^{N} w_i = 1\}, \quad (2)$$

where N is the total number of considered criteria and w_i stands for the preference weight of criterion i. The weight indicates the degree of significance of each criterion in the network selection strategy. An appropriate network is selected based on weighted user preferences and the instantaneous measured values of the three metrics. The network selection trigger is computed by:

$$Trigger = (w_p, w_e, w_c) \times \begin{pmatrix} u(x_p) \\ u(x_e) \\ u(x_c) \end{pmatrix}$$

The weight w of each metric is derived from the priority preferences configured by end-users where a bigger number indicates higher priority level, and $u(x)$ is the elementary utility function as Equation 1. We have conducted preliminary studies of user feedback based network selection [12].

This network selection heuristic scheme offers a practical way to select networks. It is also simple and lightweight without complicated input parameters . However, this threshold-based approach is sensitive to changes in devices, network environments, and user behaviors. It probably performs well in controlled environments but poorly in a different situation. We next introduce the application context model to improve its robustness and accuracy.

C. Application Context Model

A naive Bayes classifier [13] is used to compute the probability to use a certain network type. We used Bayes based on such an intuition that if people has used certain network type before, they will most likely use the same network type in the future. The Bayes classifier is a distribution defined as follows:

$$Prob(net) = \begin{cases} 1, if \quad net = \arg\max_{n_i} P(n_i|V) \\ 0, otherwise \end{cases}$$

The probability $P(n_i|V)$ is conditioned on several feature variables V, which has the following relation according to conditional probability:

$$P(n_i|V) \propto P(n_i)P(V|n_i),$$

where $P(n_i)$ is the prior probability for network type i, and $P(V|n_i)$ is the probability for network i. Here, label n_i is denoted as the network type, $n_i \in \{n_1, n_2, ..., n_m\}$. The feature V is a vector consisting of network performance, traffic volume, energy consumption, and running applications, denoted as $V =< p, c, e, app_1, app_2, ..., app_k >$.

For the first three variables (i.e., network performance p, traffic volume c, and energy consumption e), we use the multivariate probability density function for Gaussian distribution to represent the conditional probability $P(p, v, e|n_i)$, denoted as $N(\mu, \Sigma; c_i)$. Parameters only consist of the mean and covariance of the three variables. The variable v_k is a discrete value as application types, the conditional probability $P(p, c, e|n_i)$ is directly calculated from training data set using Bayes' theorem. For remaining variables, applications p_j, probability $P(p_j : j \in apps|n_i)$ are conditionally independent, given network type n_i. The probability is the product of all conditional probabilities, as $\Pi_{j\in app}P(p_j|n_i)$. The underlying assumption of a naive Bayes classifier is that, given the class variable, the presence or absence of a particular feature is unrelated to the presence or absence of any other feature. Given the network type, it is reasonable that running applications have no correlations with other applications in user preference.

The context model gives a probability with the network type. When the probability the current selected network is lower than other network types by a pre-specified *threshold*, then the context model would select the network type with the maximum probability. Otherwise, the context model will keep the original network type. However, due to sheer number of applications and uncertainty of application usage for smartphone owners, the context model gradually becomes outdated and useless. We next present an idea on crowd-sourcing applications data to deal with this issue.

D. Crowd-sourcing Applications Data

As mentioned previously, constructing an application-aware model for network switching faces with huge number of applications on Smartphone markets and uncertainty usage of applications for users. We use crowd-sourcing to collect training data for an application-aware model. However, using crowd-sourced [6] data to build a context model has its pros and cons. The merit is that soliciting similar applications from a large group of people will reduce the model training burden for individuals, specially considering the huge number of applications on the market. The drawback is that the accuracy of a context model will degrade heavily due to the diversity of people, if crowd-sourcing is not carefully done. For example, some people use smartphones very differently from others as shown in Figure 3. We propose two similarity metrics – smartphone usage similarity and geo-label similarity– to divide crowd users into different groups as shown in Figure 6. The similarity value of two persons is high in the same group. We apply a hierarchical clustering technique to identify different groups based on the similarity feature. In the same group, the application context model is trained by sharing applications data.

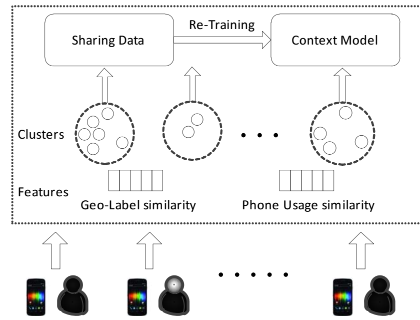


Fig. 6: Crowd-sourcing application data for the context model

1) Smartphone Usage Similarity: We compute smartphone usage similarity using three types of information: applications usage, network traffic volume, and battery during time. We utilize applications usage motivated by Figure 4, where two users have similar applications preference. Applications usage similarity is only based on the popular application categories [11] instead of specific applications usage in the naive Bayes classifier. For example, Facebook and Twitter are both considered as social network applications. We utilize network traffic volume and battery during time because they reflect network type usage preferences and energy usage preference. These two features are factors most users care about.

Collectively these three factors used as smartphone usage features can reveal the population diversity issue, hence a unified context model is not desirable. Each of these three values is an element in a vector that represents a single user smartphone usage. The similarity between users (i, j) is based on the mahalanobis distance [14] which is commonly used for calculating the distance between two discrete variables.

$$Sim_{i,j} = \exp(-\alpha(\mathbf{p}_i - \mathbf{p}_j)^\top \Sigma^{-1}(\mathbf{p}_i - \mathbf{p}_j))$$

where, $\mathbf{p}_i$ and $\mathbf{p}_j$ are the vectors for user i and user j, Σ is the covariance matrix and α is a scaling parameter empirically determined. For robust estimation of two persons smartphone usage similarity, we take into account $Sim_{i,j}$ at time t over a window of T time units:

$$Sim_{i,j}(t) = \omega_1 Sim_{i,j}(t) + \omega_2 \frac{\sum_{t_i \in [t-T,t]} Sim_{i,j}(t_i)}{|N_{t_i}|}$$

where ω_1 and ω_2 adjust the weights between the two terms of current status and the entire window history, $\omega_1 + \omega_2 = 1$. This improves overall robustness and avoids treating current similarity metric as an abrupt changing inter-persons similarity.

2) Network Conditions Similarity: We propose geo-label as the feature to calculate network conditions similarity between two persons. The reason is that network environment is correlated with locations of smartphone users. For example, networks in home environment are different from work places, leading to different network selection trends.

We do not use GPS as geo-labels due to its high energy consumption and its ineffectiveness in indoor environments. We approximate geo-labels using WiFi and cellular fingerprints as features. It is unnecessary to map cell tower fingerprints or WiFi signal strength to a specific spatial location (i.e., longitude and latitude), as directly adopting fingerprints is enough to calculate the similarity of pairwise network condition. Our similarity metrics calculation is inspired and based on RADAR [15]. For signal strength of WiFi APs, we estimate the similarity via the Euclidean distance between two vector lists since each WiFi AP has a list of signal strength vectors A_i. Similarity of WiFi signal strength between user i and user j is computed as follows.

$$Sim_{i,j} = 1 - \frac{||A_i - A_j||}{d_{max}},$$

where d_{max} is the maximum possible RSSI distance (e.g. $d_{max} = 32$). For cellular fingerprints, we estimate the similarity capturing both the number of matching cell IDs between two fingerprints and the Euclidean distance between the signal strength vectors of the matching towers. Similarity of cellular fingerprints between user i and user j is computed as follow.

$$Sim_{i,j} = \omega_e N + (d_{max} - ||C_i - C_j||),$$

where N is the number of observed cell tower IDs, ω_e is an empirical weighting parameter, and $C_i and C_j$ are signal strength vectors of cell tower. We adopt the maximum value of similarity to normalize similarity metric.

3) Clustering: We do not assume *a priori* knowledge of the number of groups in crowds, so simple clustering methods like K-means are infeasible. We use agglomerative hierarchical clustering to discriminate the crowd-sourcing data. The features of clustering are similarity values of geo-labels and smartphone usages. The hierarchical technique computes the similarity of all data in a cluster to their cluster centroid and is defined by

$$Sim(X) = \sum_{d \in X} Sim_{d,c},$$

where d is the data collected through crowd-sourcing in cluster X, and c is the centroid of cluster X, i.e., the mean of all vectors. The choice of which pair of clusters to merge is made by determining merging which pair of clusters will lead to smallest decrease in similarity. After clustering, each cluster represents a group in crowds, and an application context model is trained using shared data.

IV. Case Study

We validate context-aware handoff through one case study, handoff between two widely accessible networks: WiFi (802.11b/g) and Cellular (WCDMA). Context-aware handoff is implemented on the Android phone. We conduct the experiment in a real world scenario.

A. Implementation

We implement context-aware handoff on Android platforms (version 2.3 Gingerbread). To set up the monitoring component in Figure 5, dynamic streaming data is sampled every three seconds, including running applications, signal strength, battery states, and traffic volume. Pre-specified data for calculating network metrics is designed as static parameters using configuration files in XML. For example, the unit energy consumption of data transfer and being idle for each network type is pre-defined as prior knowledge. These pre-specified data will change when new technology or hardware is adopted in the future, and configuration files are easy to modify.

To set up the network selection component in Figure 5, a GUI is used to configure user preferences about three metrics of networks selection: performance, energy and cost. Each metric has four discrete values to calculate weights for user preference as in Equation2: 3 for high, 2 for medium, 1 for low, and 0 for do-not-care. We set steepness parameters $\zeta = 2$ to indicate user sensitivity in the utility function as shown in Equation1.

To set up application context model component in Figure 5, we train the naive Bayes model for classifying each network type based on crowd-sourced data. The crowd-sourced data is the data set of smartphone usage traces of 45 people, including four aspects: time duration between two battery charging, cellular network traffic, the ratio of cellular to WiFi usage, and application usage preferences, as mentioned in the section II-B. We include different people during different time periods for crowdsourcing data collection, since it better represents the diversity of the context model. We reuse functionalities of the Funf Open Sensing Framework [16], which is an extensible and reusable sensing and data processing framework for logging these data. Participants use their phones as usual. Crowdsourced data clustering is performed offline. We apply the WEKA tool [17] to cluster crowd-sourced data into different groups via a hierarchical clustering. In the same group, the shared application data set is used to train the naive Bayes classifier (e.g., estimated mean and variance). Parameters of a classification model are stored in Android Phones for construction of the naive Bayes classifier. We port the existing naive Bayes classifier code to Android Phones, and verify that it works correctly.

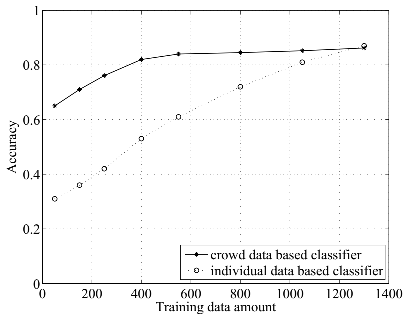


Fig. 7: Accuracy of classifier based on crowds training data and individual training data

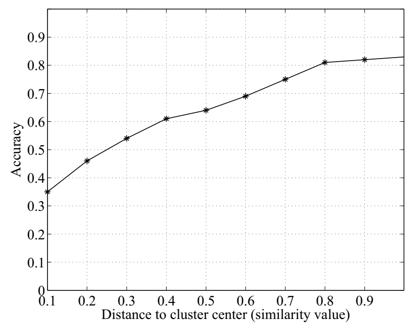


Fig. 8: Impact of distance to cluster center on the accuracy of context models

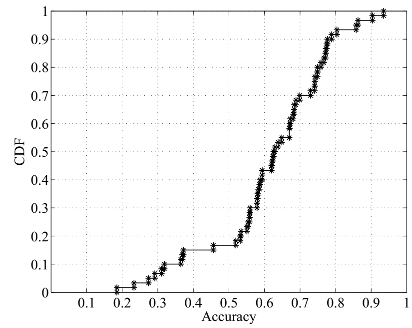


Fig. 9: Distribution of accuracy of context model in the same cluster

B. Real Deployment

To quantify the performance of context-aware handoff in a real world scenario, we conducted an experiment in the surroundings of Institute of Information Engineering at the Chinese Academy of Sciences (IIE-CAS), including an outdoor walkway with sparse WiFi APs coverage and full cellular network coverage as shown in Figure 10 (top), and an indoor environment of a two-floor building with dense WiFi APs coverage and cellular network in several corners with weak signal strength as shown in Figure 10 (bottom). We collected 10 traces, each trace with one random application running at the background. For each trace, we used three Android phones with one network selection scheme on each phone: the state-of-the-art switching mechanism on the phone (mostly adopt only signal strength), the context-aware handoff policy, and a heuristic network selection without context models. State-of-the-art switching mechanism on Android platform selects the network type in a crude way. Basically, the wireless connection manager selects WiFi as long as WiFi is available, but switches to cellular network when WiFi is not available.

In our performance studies, we first evaluated the performance of the application context model; we then compared the three network selection schemes in terms of network performance, energy consumption, and cost. The last part of the studies focuses on user experience comparison.

1) Application Context Model : As stated before, we evaluate the naive Bayes classifier for application context model because the training data comes from the crowd-sourced data rather than personalized training data set. Figure 7 uses two classifiers based on different training sources, individual and crowd training data. When the amount of training data is small, the accuracy of classifiers based on individual data is as low as 30%, unacceptable for context-aware handoff. Crowd-sourcing approach performs well even when the individual data set is small, since the data from other users in the cluster may supplement it. From Figure 7, we learn that the crowd sourcing approach is able to lower the user burden of contributing training data. We further investigate the accuracy of the classifier for different people in the same cluster as shown Figure 8 9. If we use the similarity values to represent the distance, The distance to the cluster center has a high impact on the accuracy of model. The longer the distance, the bigger difference the personal model is from the model of the cluster. As shown in Figure 8, when the similarity value is higher than 0.7, the accuracy is nearly 75%. From the CDF of accuracy of context model in the cluster (Figure 9), more than 65% of users have an accuracy between 0.55 and 0.8. From Figure 9 in conjunction with Figure 8, we learn that the context model from similar clusters can be applied for personal usage regardless of the diversity among people in crowds.

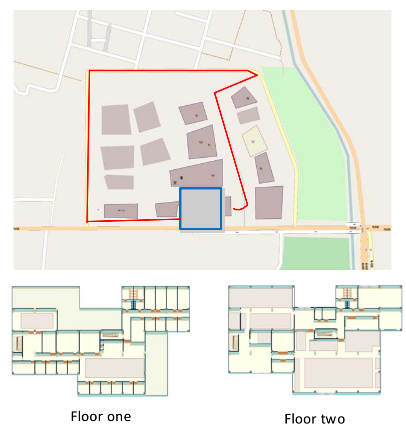
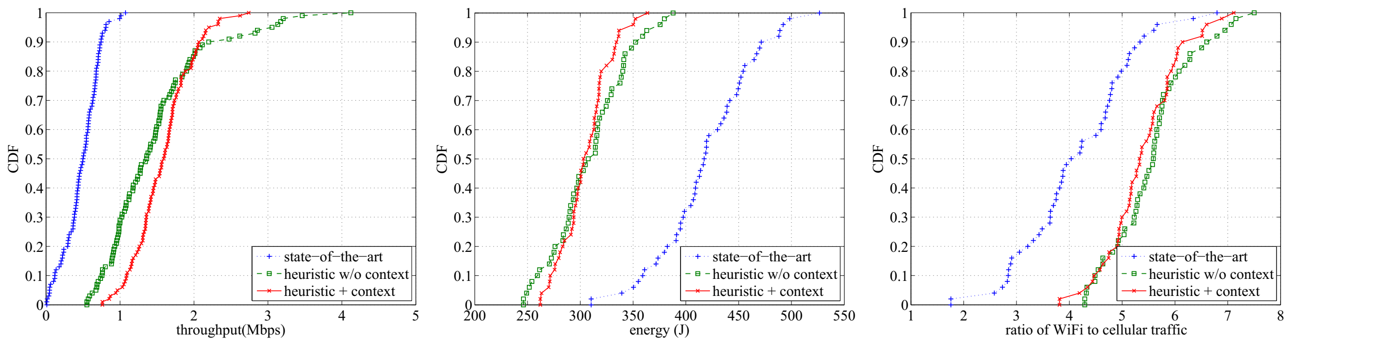


Fig. 10: Map of paths traveled outdoor (top) and indoor (bottom)

C. Evaluation

1) Network Performance: We evaluate network performance using the *iperf* tools for Android platforms installed on smartphones. Figure 11a shows the CDF of throughput under three network selection schemes. The context-aware handoff policy has more than twice throughput than the state-of-the-art switching mechanism of mobile phone (based on signal strength). Only less than 10% throughput of the context-aware handoff is below 0.3Mbps and the highest throughput is 1-2Mbps.



(a) network performance **(b) energy consumption** **(c) data offloading**

Fig. 11: Comparison of three network selection schemes

2) Energy Consumption: We evaluate energy consumption using the PowerTutor tool. Figure 11b is the energy consumption distribution of WiFi interface and cellular interface under the three network selection schemes. Average energy consumption of the context-aware handoff policy is only 25% of the energy consumption of the state-of-the-art switching mechanism of mobile phones. The reason is that when the amount of transfer data is small or the network interface is idle most of the time, WiFi consumes more energy than cellular. The state-of-the art switching scheme always keeps WiFi interface accessible even when its signal strength is week and there is little data transfer, but the heuristic scheme gives higher weight to energy as indicated by Equation 2.

3) Cost: We use data volume offloaded from cellular to WiFi network as the degree of cost, as WiFi is often free, but cellular has its owner bill. The higher the ratio of data offloading, the more savings. There are public APIs available to collect this information on current smartphones. Figure 11c is CDF of the ratio of data transfer volume from cellular network to WiFi network for all three network selection schemes. It shows that the ratio of the context-aware handoff policy mostly hovers around 5.5, and the ratio of the switching mechanism varies a lot from 0.9 to 8.3. In terms of the average amount of data offloaded from cellular network to WiFi network, context-aware selection scheme has about 40% more than the switching mechanism.

4) User Experience: As shown in Figures 11a, 11b and 11c, context-aware handoff policy does not have much improvement in terms of network performance, energy consumption of network access, and cost of network access compared with the heuristic network selection scheme without a context model. This is because the application context model is designed to improve user experience, not to improve these metrics per se. Because user experience is subjective, we use two indicators to represent: the number of network switchings and the probability of network flows disconnection during one time switching. Reasons that we choose these two metrics are that less network switching operations and lower probability of network disconnection both reflect better user experience,

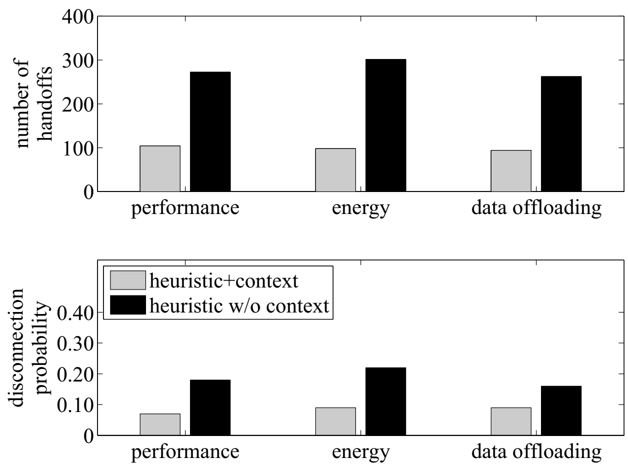


Fig. 12: Two handoff policies in three scenarios. Top: heuristic handoff and context-aware handoff; bottom: probability of network flows disconnection during one handoff.

given similar network selection criteria such as performance, energy and data offloading. Figure 12 (top) shows the total number of handoffs using the context-aware policy is only one-third of the heuristic networks scheme when network performance, energy consumption, or cost is similar. Figure 12 (bottom) shows the probability of network flows disconnection when one handoff occurs, given similar network performance, energy consumption, or cost. We observe that the network flow disconnection using the context-aware policy is only half of the heuristic network selection scheme. The reason is that the application context model determines whether network switching is suitable or not via results classified. The context-aware handoff ensures comparable network performance, energy consumption, or cost with the heuristic networks scheme, while still incurring much less handoffs.

V. Related Work

Handoff among multiple networks inevitable brings about the interruption of ongoing network flows, leading to user experience degradation. Mobile IP [18] allows mobile devices to switch between different networks while maintaining a permanent IP address in the home network. Each network flow is forwarded by home agent of devices, eliminating the impact of network interruption. Extra infrastructure such as gateway [19]–[21], acts as intermediate data forwarding proxy between devices and networks. These two approaches require either infrastructure support or network protocol modification. In contrast, our work focuses on the software level without the need to modify network protocols.

There are two pieces of work [22] [23] that eliminates network fluctuations for network switching in local phones. Seamless [22] suggests either waiting for some time or resuming network connection in order to reduce the impact of interruption on smartphones. Multinets [23] implements the network flows migration between WiFi and Cellular on the Android platform. They both focus on how to eliminate network frustrations caused by network switching in local phones. In contrast, our work is about the choice of the most suitable connection at a given time, rather than migration of network flows for one time network switching. The most appropriate choice depends on both network conditions, applications, and user preferences.

Previous work on multiple networks selection [7]–[9] chooses the most suitable network based on the assumption that QoS of applications and service was prior knowledge or queried via a database. This assumption does not hold because of popular and pervasiveness of smartphone applications. Unlike previous work, we do not assume any prior knowledge of applications.

VI. Conclusion

In this paper, we consider the problem of user experience degradation caused by switching between multi-networks while striving for network performance improvement, energy efficiency and cost conservation of network access. We first conduct a empirical study about user experience and smartphone usage related to network handoff. We then design a context-aware handoff policy for network switching in modern smartphones. A hierarchical clustering is applied to divide crowds into different groups, using features consisting of smartphone usage and geo-labels. We train the naive Bayes classifier for applications using the data set shared in the same group. We finally conduct a case study, handoff between cellular and WiFi networks, and deploy the system in a real-world scenario for evaluation. Compared with the standard network selection scheme used in modern smartphones, our context-aware handoff achieves 25% energy consumption, nearly 40% more data offloading and more than twice throughput with only one third of handoffs.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No.60933011), State Key Development Program for Basic Research of China (Grant No.2011CB302902), National High Technology Research and Development Program of China (Grant No.2012AA050804) and the Innovation Program of Institute of Information Engineering Chinese Academy of Sciences (No. Y3Z0071H02).

References

[1] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3g using wifi," in *Proc. of MobiSys, San Francisco, USA.*, March 2010.

[2] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces," in *Proc. of MobiSys, Uppsala, Sweden.*, March 2006.

[3] S. Dimatteo, P. Hui, B. Han, and V. Li, "Cellular traffic offloading through wifi networks," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 192–201.

[4] Google. (2011, Jun.) Eric schmidt at mobile world congress.

[5] Apple. (2011, Jun.) Apple's app store downloads top 10 billion. [Online]. Available: http://www.apple.com/pr/library/2011/01/22appstore.html

[6] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[7] S. Lee, K. Sriram, K. Kim, Y. H. Kim, and N. Golmie, "Vertical handoff decision algorithms for providing optimized performance in heterogeneous wireless networks," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 2, pp. 865–881, 2009.

[8] A. H. Zahran, B. Liang, and A. Saleh, "Signal threshold adaptation for vertical handoff in heterogeneous wireless networks," *Mobile Networks and Applications*, vol. 11, no. 4, pp. 625–640, 2006.

[9] S. Mohanty and I. Akyildiz, "A cross-layer (layer 2+ 3) handoff management protocol for next-generation wireless systems," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 10, pp. 1347–1360, 2006.

[10] Q. Nguyen-Vuong, N. Agoulmine, and Y. Ghamri-Doudane, "A user-centric and context-aware solution to interface management and access network selection in heterogeneous wireless environments," *Computer Networks*, vol. 52, no. 18, pp. 3358–3372, 2008.

[11] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman, "Identifying diverse usage behaviors of smartphone apps," in *Proc. of IMC, Berlin, Germany*, October 2011.

[12] Q. Li, W. Qin, L. Li, and L. Sun, "Smartphone heterogeneous network handoff based on the closed control loop," in *Proc. of ICCPS (Poster Session), Bejing, CN*, September 2012.

[13] C. M. Bishop, "Pattern recognition and machine learning (information science and statistics)," 2007.

[14] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[15] P. Bahl and V. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2000, pp. 775–784.

[16] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland, "Social fmri: Investigating and shaping social mechanisms in the real world," *Pervasive and Mobile Computing*, 2011.

[17] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[18] C. E. Perkins, "Mobile ip," *Communications Magazine, IEEE*, vol. 35, no. 5, pp. 84–99, 1997.

[19] R. Chalmers and K. Almeroth, "A mobility gateway for small-device networks," in *Proc. of PerCom.*, 2004.

[20] D. Maltz and P. Bhagwat, "Msocks: An architecture for transport layer mobility," in *INFOCOM'98.*, vol. 3. IEEE, 1998, pp. 1037–1045.

[21] I. Ramani and S. Savage, "Syncscan: practical fast handoff for 802.11 infrastructure networks," in *INFOCOM 2005*, vol. 1. IEEE, 2005, pp. 675–684.

[22] A. Rahmati, C. Shepard, C. Tossell, L. Zhong, P. Kortum, A. Nicoara, and J. Singh, "Seamless tcp migration on smartphones without network support," *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[23] S. Nirjon, A. Nicoara, C. Hsu, J. Singh, and J. Stankovic, "Multinets: Policy oriented real-time switching of wireless interfaces on mobile devices," in *Proc. of RTAS, IEEE, Bejing, CN.*, March, 2012.