An Rssi-Aware Download Management Approach for Energy Saving on Smartphones

Lunde Chen, Huan Li*, *Member, IEEE*

***Abstract*—Smartphones are emerging as a particularly appealing platform for network applications, especially through Wi-Fi interface. However, Wi-Fi entails considerable energy consumption and smartphones are bottlenecked by their battery capacity. Finding ways to reduce energy consumption of Wi-Fi is more than critical. In this letter, we address the problem of Wi-Fi energy management with the awareness of Rssi properties. Through experiments, we first investigate how Rssi influences energy consumption. Based on the experimental findings, we propose a simple but effective approach for managing large file downloading in Wi-Fi environment for smartphones. Simulations show that the proposed method can achieve significant energy saving in both moving and stationary-moving scenarios.**

***Index Terms*—Rssi, Energy saving, algorithm.**

I. Introduction

SMARTPHONES are becoming increasingly popular and have emerged as a particularly appealing platform for network applications, especially through Wi-Fi interface. However, these light-weighted and easy-to-carry mobile devices are constrained by their limited battery capacity. We consider the scenario where people use their smartphones to download large files such as movies, documents and apk files. Those files are typically larger than 10 MB and may take several seconds or even minutes to complete the download. The key challenge here is how to perform the download activity in an energy-efficient way so that it will not affect the normal usage and other applications running on the smartphone.

In this letter, we first investigate the impact of Rssi strength on the energy consumption and observe that downloading one file in weak Rssi environment can consume 8 times more energy than in strong Rssi. Hence, an ideal strategy is to download the file when the Wi-Fi signal is strong and stable; and stop the action if the signal is weak.

However, smartphones are naturally for the mobile usage, which will obviously result in a constantly changing Rssi environment. Furthermore, even the smartphone is keeping stationary, Rssi is not stable and has considerable fluctuations [1], [2].

Different from other prediction-based schemes, here, we propose a simple on-line algorithm that can automatically adjust the download actions by perceiving the varying Rssi in real-time. Simulation results show that the proposed method can effectively save the energy whenever the phone is moving or in stationary-moving status.

*Corresponding Author: Dr. H. Li (email: lihuan@buaa.edu.cn), is with the School of Computer Science and Engineering, Beihang University, Beijing, China 100191.

L. Chen is with with the School of Computer Science and Engineering and Ecole Centrale de Pékin, Beihang University, Beijing, China 100191.

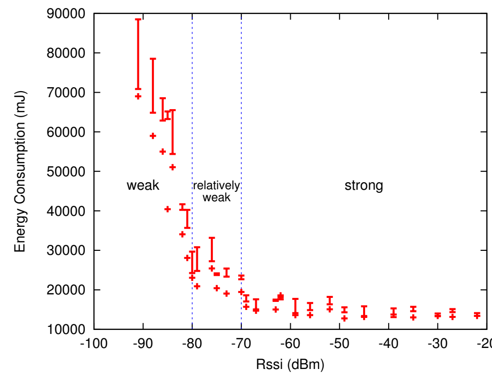

Fig. 1: Energy Consumption in different Rssi

II. Evaluation of Rssi and Energy Consumption

In this section, we first present the experiment setup. Then we discuss some interesting observations.

A. Experimental Setup for Power Measurement

We use a Monsoon Power Monitor [5] for the power measurement. The DUT (Device Under Test) is a Huawei 8950D running Android 4.0, equiped with a double-core 1.2GHz Snapdragon CPU and 768MB RAM, supporting IEEE 802.11 n/b/g. The Monsoon Power Monitor supplies a stable voltage of 3.7V to the DUT and samples the power consumption at a rate of 5KHz.

We developed an application using DownloadManager[1] to perform the download action. We cross compiled Iptables and Tcpdump and installed them into the DUT as local libraries. Iptables is used to block the Internet access of all applications except that of the download application so that no interference traffic is introduced. During the downloading time, the screen is off and the Wi-Fi keeps awake. Since a typical size of an apk file is about 10MB, we measure the power consumed for completing downloading a file of 11.4MB. At one location, each measurement is repeated three times; and then the instruments are moved to another location where Rssi is different.

B. Measurement Results

Fig.1 shows the experimental results for the energy consumption over different Rssi. We observe that: 1) when Rssi is higher than -70dBm, the energy consumption is limited and

[1]DownloadManager is a system service of Android that handles long-running HTTP downloads.

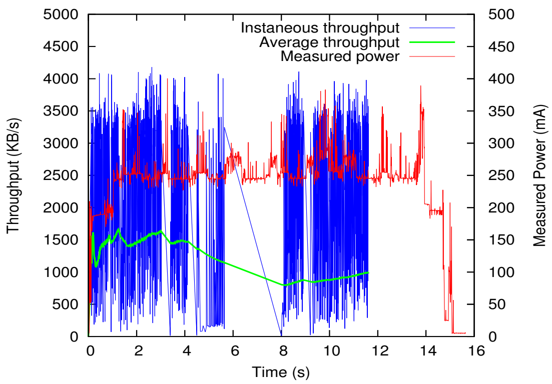

Fig. 2: Throughput vs power (Rssi = -30dBm)

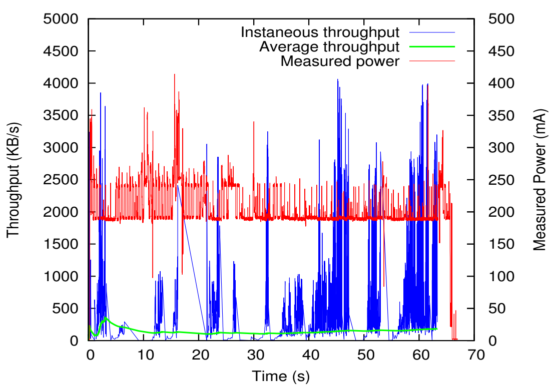

Fig. 3: Throughput vs power (Rssi = -84dBm)

independent of Rssi; 2) when Rssi is between -80dBm and -70dBm, the energy consumption increases as Rssi becomes weaker, but not that significant; 3) when Rssi is below -80dBm, the energy consumption increases dramatically. So we mark the signal strength as "strong", "relatively weak" and "weak" for those representative Rssi ranges, as illustrated in the figure.

The results for the throughput vs power for different Rssi strengths are shown in Fig.2 and Fig.3. An interesting observation is that although the power measured is slightly higher in strong Rssi than in weak Rssi, the throughput in strong Rssi is more than 5 times higher than in weak Rssi (1006KB/s vs 183KB/s). Thus, the total time for completing the download in strong Rssi is only 1/5 as in weak Rssi (11.4s vs 64s), which results in much smaller value in energy consumption (13312mJ vs 54397mJ).

Another interesting observation is that for both situations, when the download is finished, Wi-Fi interface will continue to work for 3-5 seconds. And the results keep the same for small file (in the order of 10KB and of 100KB) downloading actions. Our analysis using Tcpdump shows that the extra 3-5 seconds of energy consumption is caused by the closure of TCP connection, as illustrated in Fig.4. We conclude that downloading files via Wi-Fi introduces significant tail energy when stopping or finishing a download. Although existing researches have shown the tail energy consumption in cellular networks [6], this is the first time that the tail energy phenomenon in Wi-Fi environment is observed and investigated. Therefore, the overhead of tail energy cannot be ignored when we consider the Wi-Fi management for smartphones.

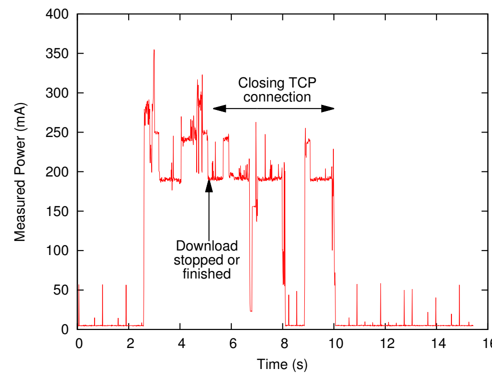

Fig. 4: Tail energy due to the closure of TCP connection when stopping or finishing a download

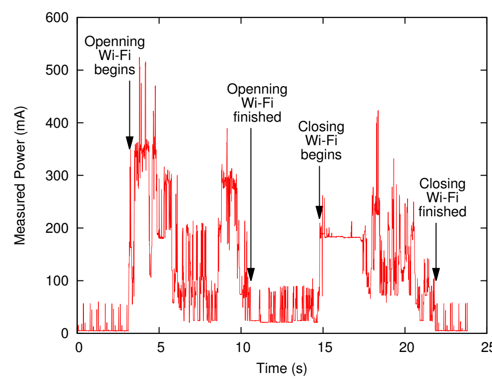

Fig. 5: Power measurement for opening and closing Wi-Fi interface

III. Rssi-aware Algorithm

As Rssi has a significant impact on the energy consumption for file downloading in Wi-Fi environment, one may consider closing Wi-Fi interface when Wi-Fi signal is becoming weak and reopening it when Wi-Fi signal is becoming strong [7]. However, our experiments show that both closing and openning Wi-Fi interface consume a lot of energy, i.e. 3700 mJ and 4300 mJ respectively, as shown in Fig.5. Keeping Wi-Fi interface open, on the other hand, consumes only 3.95 mW overhead. As the result, the energy consumption of closing and opening Wi-Fi interface is equal to that of keeping Wi-Fi interface open for 2025 seconds. Besides, keeping Wi-Fi interface open allows to monitor the Wi-Fi condition in real-time and adjust the smartphone's networking behaviors accordingly. To this end, in the algorithm design we keep Wi-Fi interface open.

We define three states to describe the current condition in which the smartphone is be aware of the Wi-Fi signal. The transition among the three states is depicted in Fig.6. The basic idea of the algorithm is to start downloading file when the state of Wi-Fi is *good*, and to stop downloading when the state is *bad*. If the state is *unknown*, no action will be taken.

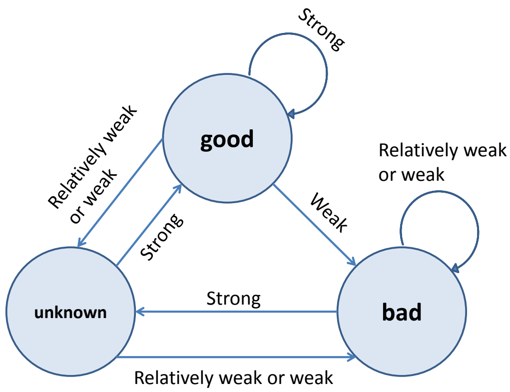

Fig. 6: Rssi-aware state transition

The algorithm keeps working repeatedly and checks the Rssi status every second, till the end of the download activity. At each iteration, the current state of Wi-Fi is updated according to the status of Rssi. The pseudo-code of the algorithm is given in Algorithm 1.

The main purpose of the algorithm is to leverage the varying nature of Rssi. As discussed before, according to its impact on energy consumption, the Rssi strength can be divided into three status: "strong", "relatively weak" and "weak". As each Rssi measurement may introduce inaccuracy, the *unknown* is set so as to leave the decision-making in the next iteration when it is not certain whether the change of Rssi is due to the smartphone user moving or due to the fluctuations of Rssi itself. This is effective because the relationship between Rssi and energy consumption is based on the threshold, as shown in Fig.1, the delay of one second in decision-making is tolerable for Rssi detection and Wi-Fi state transition. In this way, we can avoid the constant "start" and "stop" downloading actions.

Algorithm 1 Rssi-based Download Management Algorithm

```
while true do
    /*Update current state of Wi-Fi,
    *as shown in Fig. 6 */
    curr_state ← update_state(prev_state, rssi_measured);
    if curr_state is good then
        start_download();
    else if curr_state is bad then
        stop_download();
    end if
    prev_state ← curr_state;
    sleep(); //sleep for 1 second
end while
```

The algorithm is light-weighted and each decision-making takes O(1) computation. Unlike other methods that may incur extra energy consumption for the complicated prediction computation, the energy overhead of this approach is negligible during the running time, since only the Wi-Fi state is stored and updated according to the current Rssi strength.

We implemented our algorithm as an application. The energy overhead of running the application is too insignificant, far less than 0.001mA. Therefore, the energy overhead of the algorithm can be assumed to be zero.

IV. Simulation Evaluation

In this section, we present the Monte Carlo simulation of the Rssi-aware Wi-Fi download management approach and the simulation results.

A. Simulation Methodology

User's Moving Behavior Model: We model the moving behavior of a smartphone user as a directional random walk [**?**], which is defined as:

$$s_{k_t} \sim \mathrm{U}(s_{min}, s_{max}) \tag{1}$$

$$\theta_{k_t} \sim \mathrm{N}(\theta_{k_{t-1}}, \sigma^2) \tag{2}$$

where s_{k_t} and θ_{k_t} are the speed and the heading of the smartphone user at instant t.

To simulate all possible starting point that the user could take, a 2-dimensional Halton sequence [8] with base {2, 3} is generated. When the user gets out of the signal range of Wi-Fi AP, we consider one walk is terminated and take the next value of the Halton sequence as his new starting point. This process is characterized as Equation (4), where R refers to the radius of the signal range of Wi-Fi AP:

$$\vec{P}_{k_{t=0}} = R\mathrm{H}_k(2,3) \tag{4}$$

Considering that the user may use his phone not only on move, but also when he is in stationary status, we introduce a Markov chain to characterize this stationary-moving transition, with two state spaces *moving* and *stationary*, represented by $\delta = 1$ and $\delta = 0$. We denote p_m as the probability for the user to keep moving at the next instant if he is on move at current instant, and p_s as the probability for the user to stay still if he is in stationary status at current instant. Then the position of the user at next instant can be given as:

$$\vec{P}_{k_{t+1}} = \vec{P}_{k_t} + \delta_{k_{t+1}}(p_m, p_s) s_{k_t} \begin{pmatrix} \cos\theta_{k_t} \\ \sin\theta_{k_t} \end{pmatrix} \tag{5}$$

Rssi Variation Model: We adopt the logarithmic wireless path model [9] to calculate the real Rssi of Wi-Fi from the distance between the AP and the smartphone:

$$Rssi(d) = Rssi(d_0) + 10\eta \log(d/d_0) + \xi \tag{4}$$

where d_0 is the reference distance, typically 1 meter, η is the path attenuation factor and ξ is the shadowing factor. In addition, as each measurement of Rssi could introduce some noise, we define Equations (5)-(7) to describe the variations of Rssi in the measurements:

$$v_{k_t} \sim \mathrm{N}(0, \phi^2) \tag{5}$$

$$\Delta_{k_t} \equiv \lfloor \| v_{k_t} \| \rfloor \pmod M \tag{6}$$

$$Rssi_{measured\_k_t}(d) = Rssi_{k_t}(d) + \mathrm{sign}(v_{k_t})\Delta_{k_t} \tag{7}$$

where M refers to the maximum variation of the measured Rssi and is given as:

$$M = \lfloor \| \frac{Rssi_{k_t}(d)}{10} \| \rfloor - 1 \tag{8}$$

Monte-Carlo Simulation: In the simulation, we assume

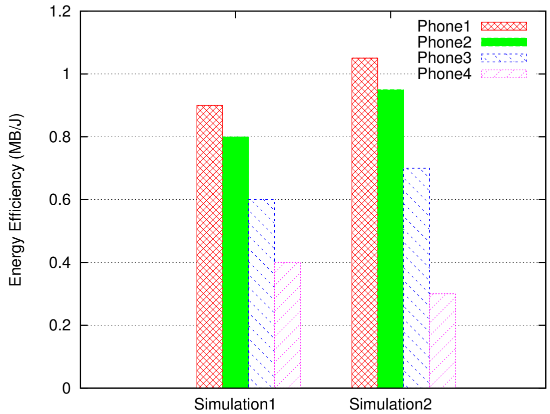

Fig. 7: Energy efficiency of the four phones

that a smartphone user is carrying 4 phones with him and performs 1,000,000 directional random walks in a circle of radius of 60 meter with an AP in the center. Phone1 runs the proposed algorithm for the download management. Phone2 starts the download if Rssi is measured to be strong and stops the download if Rssi is measured to be weak or relatively weak. Phone3 starts the download if Rssi is measured to be strong or relatively weak and stops the download if Rssi is measured to be weak. Phone4 has no download strategy and keeps downloading regardless of Rssi. The four phones are all assumed to be fully charged with infinite-capacity battery. Also we assume that the file to be downloaded is of infinite size.

Two simulations are carried out. In the first one, the user always keeps walking and thus $p_m = 1$, $p_s = 0$. In the second simulation, the user walks and stops, with $p_m = 0.8$, $p_s = 0.9$.

B. Simulation Results

We first evaluate the energy efficiency for the four phones and the results are illustrated in Fig.7. The energy efficiency is defined as the fraction of the size of file downloaded and the energy consumed:

$$Energy\_Efficiency = \frac{Size\_Of\_File\_Downloaded}{Energy\_Consumed} \quad (9)$$

For both scenarios, we found that: 1) the phones with Rssi-aware Wi-Fi management strategies can achieve better performance than the one that has no any Wi-Fi energy management approach; 2) Phone1 always outperforms the other phones; 3) Phone1 can achieve more than 100% energy saving than Phone4.

What is download session? Lunde: revise this paragraph! In order to evaluate the influence of tail energ, we also ...... With regard to the download sessions, we found that in Simulation1, Phone2 has 13.5% more download sessions than Phone1. This rate increases to 18.8% in Simulation2. More download sessions mean more actions of starting or stopping downloading files, thus wasting energy consumption in closing (and reestablishing) TCP connections. At the same time, in Simulation1, Phone1 also has 14.7% longer average download session length than Phone2. In Simulation2 this rate is 21.4%. These two factors contribute to the better performance of Phone1 in preference to Phone2.

In summary, we conclude that the proposed algorithm can successfully be aware of the changes of Rssi, and is effective in leveraging the variations of Rssi to achieve energy saving for large file downloading in Wi-Fi environment.

V. Conclusions and Future Work

In this letter, we first investigated the relationship between Rssi and energy consumption for large file download on smartphones, and then proposed an Rssi-aware management algorithm for energy saving. Through Monte-Carlo simulations, we demonstrated that the algorithm is effective since it takes both the Rssi variation nature and the Wi-Fi tail energy property in the design.

In the future, ...**Lunde: update**

Acknowledgment

This work is supported by the National Nature Science Foundation of China, NSFC (Grant No. 61170293) and the International Science & Technology Cooperation Program of China, Ministry of Science and Technology of China (Grant No. 2014DFG12370).

References

[1] Noah Pritt, *Indoor Positioning with Maximum Likelihood Classification of Wi-Fi Signals*, SENSORS 2013 IEEE, November 2013, p1-p4
[2] Doug Rosener, *Decreasing Rssi Setting Time in Low-power Mode Systems*, US Patent, Pub.No: US 2013/0029607 A1, Pub.Date: Jan 31, 2013
[3] Frank Vanheel et al., *Automated linear regression tools improve RSSI WSN localization in multipath indoor environment*, Journal on Wireless Communications and Networking, 2011
[4] Jenq-Shiou Leu, Nguyen Hai Tung, and Chun-Yao Liu, *Non-Parametric RSS Prediction Based Energy Saving Scheme for Moving Smartphones*, IEEE Transactions on Computers, July 2014, p1793-p1801
[5] Monsoon Power Monitor, *www.msoon.com/LabEquipment/PowerMonitor*
[6] Di Zhang et al., *Leveraging the Tail Time for Saving Energy in Cellular Networks*, IEEE Transactions on Mobile Computing, July 2014, p1536-p1549
[7] Jian Li et al., *Application-Centric Wi-Fi Energy Management on Smart Phone*, Network Operations and Management Symposium (APNOMS), 2012 14th Asia-Pacific
[8] Halton Sequence, *en.wikipedia.org/wiki/Halton_sequence*
[9] Lashkari, A. H., Parhizkar, B., & Ngan. *WIFI-based indoor positioning system*, Proceedings of the IEEE ICCNT 2010, 23–25 April, 2010, p.76–78