EAI Endorsed Transactions
on Mobile Communications and Applications

Research Article **EAI.EU**

Power Optimization of WiFi Networks based on RSSI-awareness

Bo Chen[1,*], Xi Li[1], Xuehai Zhou[1]

[1]University of Science and Technology of China, Hefei 215123, China

Abstract

This research analyzed the impact of Wi-Fi received signal strength indication (RSSI) on power consumption of smart phones under different network environments. It was found that bad signal may lead to decrease in network link speed but increase in power consumption; whereas good signal contributed to rapid transmission and low power consumption. To reduce the power consumption and prolong the battery life, through combination with the original IEEE 802.11 PS mechanism, an optimization mechanism based on RSSI-awareness was proposed which aggregated network packets and delayed transmission. In essence, the proposed mechanism functioned through decreasing the number of mode switches of Wi-Fi component and extending the time of Wi-Fi stay in the Power-Save (PS) mode. Specifically, the signal strength was divided into three levels, including "good", "weak" or "bad". The decision tree was used to choose the best transmission method (Normal / Lineral / Exporational Transmission) according to previous and current signal strength. Algorithms were introduced to select the best split of decision tree for partitioning specific records into smaller subsets. Finally, the proposed mechanism was performed on the TI pandaboard platform. The results indicated that the proposed mechanism was practical and able to reduce the power consumption of smartphones in an effective manner.

Received on 08 October 2018; accepted on 05 December 2018; published on 31 January 2019

Keywords: Android, WiFi, received signal strength indication (RSSI),Power Save Mode.

doi:10.4108/eai.14-12-2018.159335

1. Introduction

Smartphones have experienced tremendous and continuous growth in recent years, accelerating the development both in software and hardware. However, they encounter some limitations in power supply, which becomes a major factor interfering with the user experience. There are quite a few factors impacting on power consumption, among which signal strength is an important one. As an example, when opening the browser on a high-speed rail, or sending a message at a place far away from Access Point (AP), we often feel the phone gradually gets hotter. This can be explained by the bad signal resulting in slow link speed. More specifically, transmission for the same amount of network packets takes more power consumption in the environment with bad signal.

When designing the network protocol and architecture initially, engineers always pursued the performance factor or improvement of the network throughput, considering less or none on the power factor. Therefore, for wireless communication on smartphones, the factors including performance, power consumption and user experience must be all taken into account. Thus, how to balance those factors is the key issue to prolong the battery life of smartphones.

This paper proposed an improved method for delayed sending network packets under different network environments. Combined with the IEEE 802.11 Power-Save (PS) mode mechanism and a buffering queue, the proposed mechanism controlled the receiving and transmission of network packets based on the RSSI of Wi-Fi network.

In general, the main contributions are shown below:

---

*Corresponding author. Email: chenbo2008@ustc.edu.cn

- We analyzed the impact of Wi-Fi RSSI on energy consumption of smartphones under different network environments. In brief, weak or bad Wi-Fi signal was found to cause more power consumption; while good Wi-Fi signal may lead to low power consumption.

- To reduce the power consumption of smartphones in different network environments, by combining the original IEEE 802.11 PS mechanism, we proposed the optimization method that could recognize the dynamic network environment and dynamically adjust the Wi-Fi transmission method based on the decision tree. Essentially, the proposed mechanism took full advantage of the PS mechanism to improve battery utilization.

- We implemented the proposed mechanism on OMAP4 pandaboard ES. Many related experiments were also carried out. The experimental data showed that the optimization mechanism could not only prolong the battery life for common application under different network environments, but also improve the user experience to some extent.

2. Background and related work

IEEE 802.11 gives two operation modes for power management [3] [4]: Continuous Active (CA) mode and Power Save (PS) Mode. Some details on these two modes are given in this section.

2.1. Continuous Active (CA) and Power Save (PS) mode

For the wireless network, CA mode (left in Fig.1) is intended for devices when power is not an issue. This mode provides the best connectivity from the user perspective. In other words, it keeps the radio powered up continuously to ensure minimal lag in response time. Specifically, Wi-Fi Network Interface Control (WNIC) provides users with more high-quality network communications services. The weakness of this mode is high power consumption. For instance, it may consume 1000mW for sending network packets by WNIC, 750mW for receiving packets, and even 650mW in idle state. In this way, WNIC quickly drains out battery [5]. This CA mode consumes the most power but offers the highest throughput. In comparison, PS mode allows the network to switch to low-power state after a period of time, thereby saving energy. The AP periodically sends a frame to notify smartphones whether or not data arrive in AP, and periodically awakens the smartphones to receive the AP-sent frame and detect whether there are data to be received.

2.2. Dynamic Power Save mode and Static Power Save mode

There also have two implementations of Power Save Mode (right in Figure 1), static PSM and Dynamic PSM, respectively. Static PSM was first standardized. With Static PSM, it allows the client to sleep as much as possible by indicating the client to fall asleep immediately if the AP does not have any packet for it. Station (STA) enters PS mode by sending a Null frame to the AP with the Power Management bit set. From then on, the AP stores all packets destined to the STA in a queue and set the TIM field in the beacon frame, indicating that packets destined for the STA have been queued at the AP. And STA is woken up by every Listen interval to receive the beacon frame. When detecting that the TIM field for it has been set, it sends a PS-Poll frame to the AP. In response, AP sends the first queued frame to the STA. After receiving the queued data frame, STA sends another PS-Poll frame to the AP if the More Data field in this frame. STA continues to send PS-Poll frames to receive all the queued frames; when none is left, it goes back to sleep until the next Listen Interval. However, this process leads to long network latency and low network efficiency as a separate PS-Poll message is required for every packet. Another feature of IEEE 802.11 is dynamic Power Save. It allows the wireless device to dynamically go into power save mode if the device is associated. No network traffic is going through the card for a certain period of time, with a default value set to 100 ms. To enable dynamic power save, the wireless driver must support power save mode of operation. The dynamic power save timeout (the time for no traffic through the card before going to PS mode) is affected by the PM-QOS network latency setting. To enable power save, a wireless device sleeps at most the duration of the Delivery Traffic Indication Map (DTIM) interval. It will wake up during that period to listen for beacons from the AP, so as to find out whether the AP has buffered frames for it. Refer to The Power Savings Guide for more details as to how this works. In sum, Static PSM allows Wi-Fi to switch sleep mode immediately after packets are received. Also, Wi-Fi components will stay in sleep mode for higher energy optimization, or in active mode longer by Dynamic PSM for better performance.

2.3. Related Works

In the last few years, there has been a growing interest in studying energy consumption in mobile devices and green computation. These works mainly include: Power consumption on WI-FI/4G/bluetooth/screen components [6][7][8], power consumption profiling based on android application execution process [9][10], network protocol power optimization [11], etc. Niranjan [12]

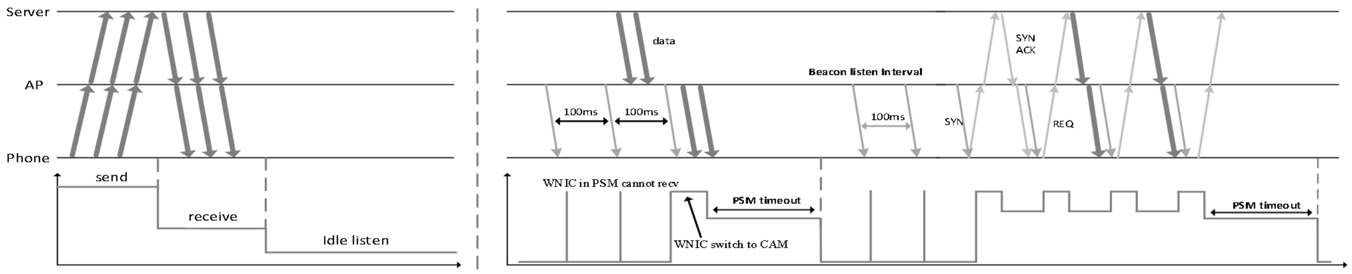

Figure 1. Continuous Awake Mode(left) and Power Save Mode(right)

conducted measurement study on the energy consumption characteristics of three widespread mobile networking technologies: 3G, 4G and Wi-Fi. He found that 3G and GSM incurred a high tail energy overhead because of lingering in high power states after completing a transfer. Hid work presented a protocol of TailEnder to reduce energy consumption. Andrew [13] proposed a novel Power Save Mode that labelled each application with a priority. High priority application would switch to CAM or Active mode, while low priority traffic was optimized for energy efficiency. Zhao [14] put forward a method focusing on web browsing. It firstly recognized the computation sequence of the web browser, and then predicted usersąŕ reading time through a practical data mining methods. It switched to low power states when the reading time was longer than a threads. Chulhong [15] developed a system which provided users with power use of sensing apps at pre-installation time. Chen [16] proposed an energy saving method for large file downloading on smartphones. In the industrial sectors, Apple MAC OS provided a service, named APP Nap. It reduced power consumption by completely suspending your appąŕs execution when certain criteria were met. The system knew a certain app that was in the background and completely hidden by other appsąŕ windows. When that app had no task, OS X would slow the app down, keeping it from using up CPU cycles, thereby saving battery power. All those works provide good references for our research.

3. Impact of RSSI on Energy Consumption

This section analyzes the impact of Wi-Fi RSSI on power consumption of smartphones. First, the communication behaviors of different apps were described. It was found that small amounts of network packets would lead to frequent switch of Wi-Fi network from PS mode to CA mode, even if a small packet was transmitted. Second, the impact of Wi-Fi RSSI on power consumption was explored, so as to make full use of this feature for transmission optimization. An experiment was performed on a TI Pandaboard ES platform [17] that carried on the network communication through WL1271 chips [18]. To change the network environment, we chose different positions from AP to smartphone. This would help to simulate the real network environment and measure the various value of RSSI. For our purpose, the experiment was designed to evaluate two factors:

(i) The network transmission behavior of different apps.

(ii) The transmission behavior of Wi-Fi component under different network environments.

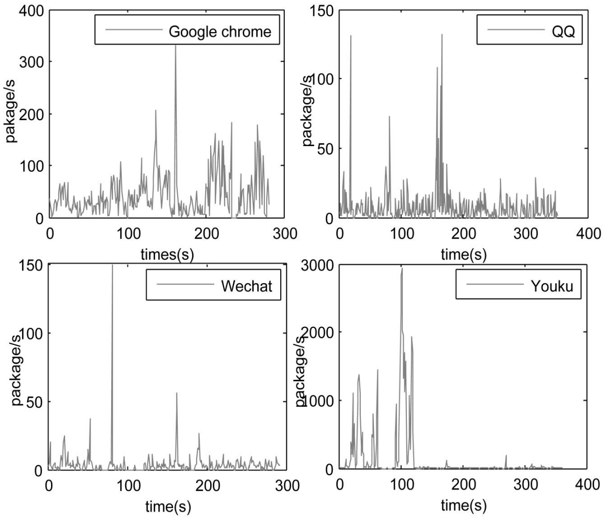

Figure 2. The communication behaviors of Different apps

For the first, as seen from Fig.2, network packets of Wechat and Tencent QQ apps are small; while those of Youku and Chrome apps are relatively big. That is to say, social apps always send small packets periodicity or sporadically, resulting in frequent switches of Wi-Fi components from CA mode to PS mode (Fig.3). Accordingly, this will increase the power consumption of smartphones to some extent.

For the second, to measure the impact of different network environments on signal strength, we ran some applications to present a conventional network transmission behavior. Games, browsers, videos and other typical applications were selected for user experience respectively.

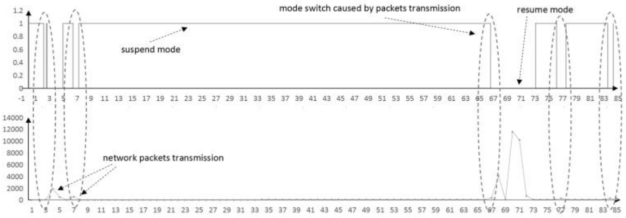

Figure 3. Network transmission behavior impact on suspend/resume mode switch

As shown in Fig.4(top), when the signal was good (RSSI between -60dBm and -40dBm) and Wi-Fi was establishing connection to AP, a relatively large jitter was produced. After successful connection, RSSI signal strength got stabilized. When the smartphone was far away from the AP, as presented in Fig.4(down), the signal was bad (RSSI below -80dBm), while the network communication condition became worse.

The average power consumption over different RSSIs was obtained as in Fig.5.

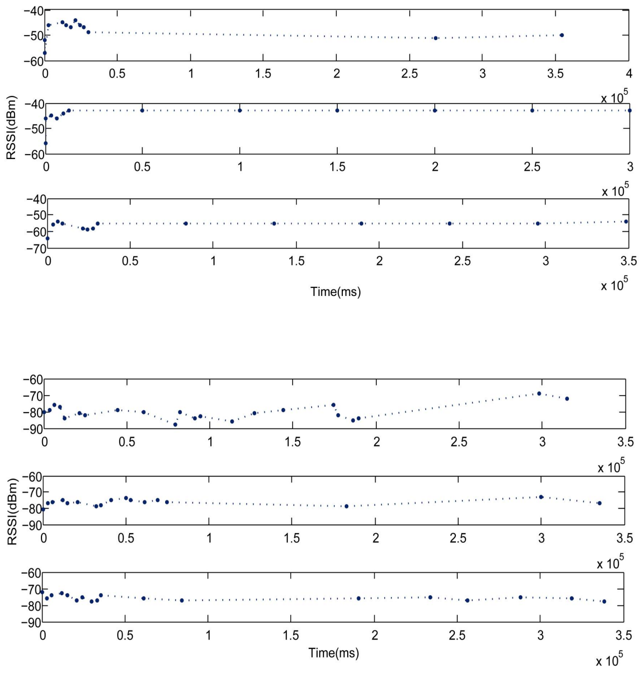

Figure 4. (Top) The jitter of Wi-Fi signal when the smartphone was near to the AP; (Down)The jitter of Wi-Fi signal when the smartphone was far away from AP

It was concluded that: 1) at RSSI above -60dBm, the energy consumption was subject to a certain limit and independent to RSSI; 2) at RSSI between -60dBm and -80dBm, the power consumption insignificantly increased with decline in RSSI; 3) at RSSI below -80dBm, the power consumption underwent a dramatic growth. Thus, the signal strength was divided as "good", "weak" and "bad", respectively, to represent different RSSI ranges.

Table 1. Statistics for different network environments

RSSI Quality	Signal
Good	Above -60 dBm
Weak	Between -60 and -80dBm
Bad	Below -80 dBm

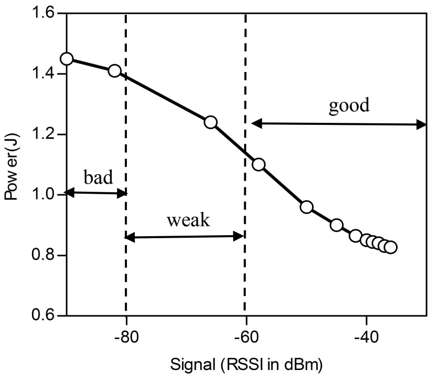

Figure 5. The average power consumption over different RSSIs

4. Architecture Design

In previous sections, we analyzed the impacts of Wi-Fi RSSI on power consumption under different network environments. To reduce power consumption, methods were proposed based on RSSI to buffer packets for different intervals. The architecture is illustrated in Fig.6.

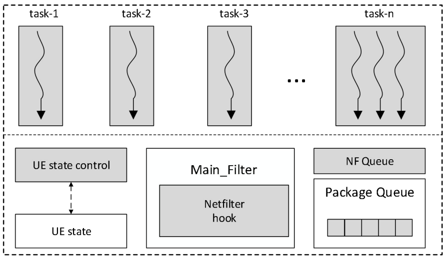

Figure 6. The framework of the optimization mechanism

The framework mainly consisted of three parts: Main_Filter, UE state control and NF Queue. The Main_Filter thread invoked the callback function in Netfilter framework [19] to buffer network packets, and executed the fork() function to create other threads. Buffered packets were controlled by NF Queue thread using Wait_Queue(), which was an internal mechanism of Linux OS used to wait for timer-based event occurrences. The UE state thread continued to track and detect the RSSI of the UE, and transferred the value of RSSI to other threads [6].

As the optimized design methods, decision tree algorithms were employed to select the best way of transmission.

4.1. Decision Tree building

Specifically, the signal strength was divided as "good", "weak" and "bad". Through the decision tree, the best way to transmission (Normal/Lineral/ Exporational Transmission) was selected on basis of previous and current signal strengths. The basic strategies for optimization were presented first. Then, impurity function was applied to construction of the decision tree.

Basic optimization operation Using Decision Tree. The principle of the optimization framework is described in the followings(Fig.7). When the signal is weak or bad, the transmission will be delayed under weak or bad signal, and be normal under good signal. For instance, when current transmission was normal, then previous status of Wi-Fi network was good (path-1 or path-4); and if current status was weak, then current path would be path-2. If the next status was good, then the total sequence of status was 'good-weak(2)-good(4)'. This situation was similar to the jitter of Wi-Fi in a place without being moved. When the next status was bad and the total sequence of status was 'good-weak(2)-bad(6)', the Wi-Fi would be transmitted by 'normal->liner delay->exponential delay'. This was similar to the situation when someone gradually got far away from the router.

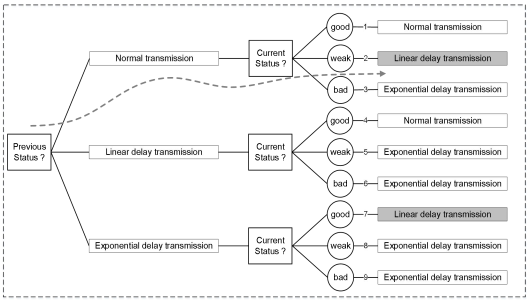

Figure 7. The Decision Tree of the framework

Implementation of Decision Tree Classification. To build the decision tree, we evaluated the different attributes using information gain theory. The greater the information gain value, the greater the purity of information obtained by partitioning with such attributes. Specifically, mainly the entropy schemes were considered in this research.

The *Entropy(D)* was defined as:

$$Entropy(D) = \sum_{n}^{k-1} P_k log_2 P_k$$

Where n represents the number of the data set; Pk represents the proportion of k classes in the data set.

The information gain of Gain(D,a) was defined as:

$$Gain(D, d) = Entropy(D) - \sum_{K=1}^{n} \frac{|D^k|}{D} Entropy(D^k)$$

The Gini(D) was defined as:

The information gain of Gini_index(D, a) was defined as:

$$Gini(D) = \sum_{k=1}^{n} \sum_{k' \neq = k} P_k P_{k'} = 1 - \sum_{k=1}^{n} P_k^2$$

The information gain of Gini_index(D, a) was defined as:

$$Gain\_index(D, a) = \sum_{k-1}^{n} \frac{|D^k|}{D} Gini(D^k)$$

$$\begin{aligned} Gain(D, "OLD\_TRAN") = Entropy(D) - \\ \sum_{K=1}^{3} \frac{|D^k|}{D} Entropy(D^k) \qquad (1) \\ = 0.064 \end{aligned}$$

$$\begin{aligned} Gain(D, "NOW\_STATUS") = Entropy(D) - \\ \sum_{K=1}^{3} \frac{|D^k|}{D} Entropy(D^k) \qquad (2) \\ = 0.124 \end{aligned}$$

The experimental data from Table 2 were utilized to illustrate the process of building the decision tree. For explanation, OLD_TRAN was adopted to describe the transmission mode of the last transmission. NOW_STATUS represents the current Wi-Fiąŕs RSSI status; NOW_TRAN stands for the current transmission mode.

For example, the attribute "OLD_TRAN" had three values of NT, LT, ET when used to divide the decision node. At first, its three branch nodes were obtained:

- D^1(OLD_TRAN=NT), it includes 30 samples(13 for NT, 10 for LT, 7 for ET) ;
- D^2(OLD_TRAN=LT), it includes 30 samples(10 for NT, 6 for LT, 14 for ET) ;
- D^3(OLD_TRAN=ET), it includes 30 samples(0 for NT, 15 for LT, 15 for ET) ;

Then, the information entropy was calculated respectively:

Table 2. Some experimental data ('NT' = Normal Transmission,'LT' = Lineral Transmission, 'ET' = Exporational Transmission)

OLD_TRAN	NOW_STATUS	NOW_TRAN
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
LT	Weak	ET
ET	Weak	ET
ET	Weak	ET
ET	Weak	ET
ET	Weak	ET
ET	Weak	ET
ET	Weak	ET
NT	Bad	ET
NT	Bad	ET

$$\text{Entropy}(\text{OLD\_TRAN}=\text{NT})=0.464$$
$$\text{Entropy}(\text{OLD\_TRAN}=\text{LT})=0.453$$
$$\text{Entropy}(\text{OLD\_TRAN}=\text{ET})=0.301$$

The information gains of attribute 'OLD_TRAN' and 'NOW_STATUS' were computed as:

The attribute 'NOW_STATUS' partition was found better than the partition of 'OLD_TRAN'. Finally, the decision tree was obtained as shown in Fig.8.

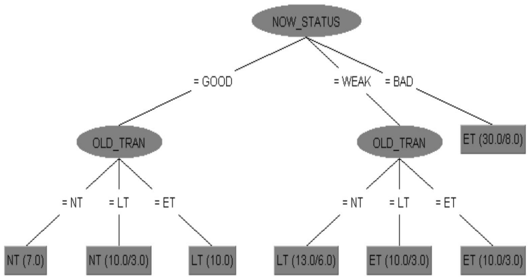

Figure 8. The decision tree obtained from the experiment

Implementation of delay-transmission mechanism. Signal strength imposes impacts on the power consumption of Wi-Fi components in many ways. The experimental results demonstrated that power consumption varied when the same application was running under different network environments. The power consumption by good signal strength was found 20% lower than weak signal strength and 50% lower than bad signal strength (Fig. 5). Most importantly, under the bad network environment, the smartphone led to delay in data reception and prolongation of the duration of awake mode. Thus, an energy optimization mechanism based on RSSI-awareness was put forward for Wi-Fi network. This mechanism was able to determine whether the current Wi-Fi RSSI was good, weak or bad, and to identify delayed-transmit packets for different intervals.

For the underlying implementation, based on the decision tree and combination with the original IEEE 802.11 Power Save mechanism, mainly three parameters (Listen Interval, Timeout, TX delay) of Wi-Fi components were adjusted to achieve delayed transmission.

- Listen Interval

The listen interval parameter specifies a number of beacon intervals. These intervals can pass from a time that a beacon frame is listened by the wireless client device from the access point before a next beacon frame is listened (shown in Fig. 9). The value of listen interval defined in WL1271 chipset ranges from 0 to 255 and can be configured through the Wi-Fi driver interface. With the good signal strength, in order to improve the response capability of the application, the listen Interval was set to a small value. With the weak signal strength, the listen interval was properly increased linearly. Similarly, listen interval was adjusted in more aggressive growth for bad signal strength exponentially.

Algorithm 1 Listen_interval control algorithm

```
if activity_stat < 60 then
    Listen_Interval_v=0;
else
    if activity_stat > 60 and activity_stat < 80 then
        Listen_Interval_v = 1;
    else
        if activity_stat > 80 then
            Listen_Interval_v=0;
        end if
    end if
end if
//when the becaon frame is received
if activity_stat < 60 then
    Listen_Interval_v + =0;
else
    if activity_stat > 60 and activity_stat < 80 then
        Listen_Interval_v + = 1;
    else
        if activity_stat > 80 then
            Listen_Interval_v * =0;
        end if
    end if
end if
if Listen_Interval_v > MAX_INTERVAL then
    Listen_Interval_v = MAX_INTERVAL;
end if
```

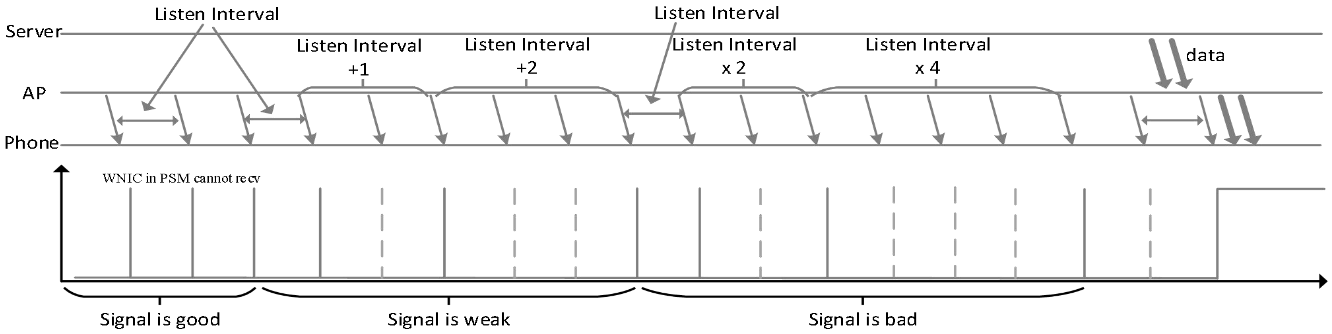

Figure 9. The implementation of delay transmission using Listen Interval

- PSM Timeout

The corresponding communication flow of mobile network communication is described as follows. First the smartphone sends a request to AP, which forwards the request packets to the server. After processing the request, the server immediately returns the response data to the smartphone. Many experiments were carried out to measure the Round Trip Time (RTT) under different network environments. As shown by Experimental Cumulative Distribution Function (CDF) in Fig. 10, the response returned to the smartphone within 100 ms.

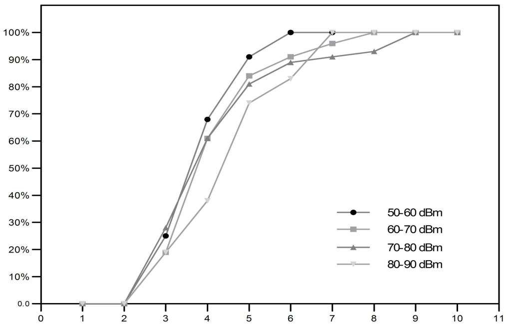

Figure 10. The experimental CDF of RTT

In the IEEE 802.11 PS mode, the parameter timeout refers to a passed time without traffic before the CA mode switches to the PS mode. As described in section 2, when a request is sent, the response data usually arrive within 100ms. Therefore timeout was set as 100ms. In this case, all the response data could be received without the need to turn to the PS mode. The periodic wake-up time increased or decreased as appropriate after 100ms to some extent. The Timeout was then set 100ms for the three RSSI conditions, so that after a request, it could quickly receive response data and avoid ąřtail powerąś.

- TX_Delay

Algorithm 2 when data transmission is completed

```
queue_work_filter(rx_streaming_enabled);
queue_delayed_work_filter(rx_streaming_disabled,
100ms);
queue_work_filter(wl1271_sleep);
```

In the IEEE 802.11 PS mode, TX_Delay represents the delay time before data is sent. In other words, it is the expression of the gathered packets to send. Buffer for sending packets in the Wi-Fi network prolongs the time of the PS mode. The prolonged time in PS mode leads to significant reduction in power consumption. Based on different RSSIs, the proposed mechanism was optimized by setting TX_Delay to various empirical values (0s/3s/5s), thereby simulating the real network environment.

Algorithm 3 when data transfer is completed

```
define BG_CACHED_TIME(3*HZ)
define SLEEP_CACHED_TIME(5*HZ)
if activity_stat < 60 then
    queue_delay_work(tx_work, 0);
    next_send_time = 0;
else
    if activity_stat > 60 and activity_stat < 80 then
        queue_delayed_work(tx_work,
BG_CACHED_TIME);
        return;;
    end if
end if
if activity_stat > 80 then
    queue_delayed_work(tx_work,
SLEEP_CACHED_TIME);
    return;
end if
```

5. Evaluation

To verify the power consumption improvement of the proposed mechanism, we performed the mechanism on OMAP4 Pandaboard ES platform. A number of experiments and tests were also conducted in different environments. Results showed that the improved mechanism contributed to effective reduction in power consumption under weak or bad signal strength, as well as better user experience under good signal strength.

5.1. experimental methods

During verification, different RSSI ranges were obtained through adjustment of the distance from mobile client to AP. To download large and small files in the case of RSSI being constantly changing, the proposed mechanism slowed down or accelerated network transmission speed to save energy consumption and improve user experience. Furthermore, an App was designed for statistical analysis on the energy consumption of different hardware components (CPU/Wi-Fi/Bluetooth/...) in smartphones. The power consumption of Wi-Fi was calculated with a power model. All statistical data were displayed in the interface.

Table 3. WL1271 Wireless NIC Parameters

Mode	Power Comsumption
Transmit mode(54Mbps)	185mA
Receive mode	100mA
Active Mode	70mA
Idle Mode	< 1.2mA

The power consumption in wireless network interface card (NIC) was calculated via the power consumption model proposed by Ning Ding of Purdue University [5]:

$$E_{total} = P_{tx} \times T_{tx} + P_{rx} \times T_{rx} + P_{active} \times T_{active} + P_{tail} \times T_{tail}$$

Where P_{tx} is unit power consumption on packets transmission; T_{tx} is the time used in transmission; P_{active} is the power spent on active mode when there is no data transfer; P_{tail} is the tail power, put simply, equal to P_{active}. P_{rx} and T_{rx}, as mentioned above, describe the power consumption of the receiving process. The power consumption was calculated under four states. The time and basic power consumption of the model at each state were analyzed, and the power consumption of each state was obtained by the product of the two data. The power consumption of the unit through the corresponding hardware datasheet was obtained as show in table 3.

5.2. experimental results

Lots of performance evaluation experiments were carried out for comparison of the proposed and original mechanisms. Different types of applications (news/video/ games /..., etc.) representing various behaviors of network communication were tested.

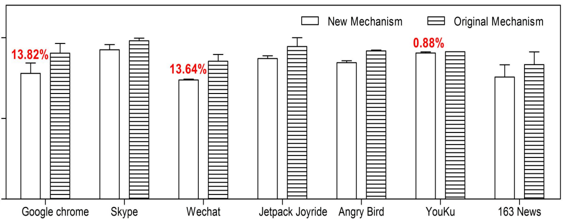

Figure 11. The comparison of Power Consumption for different Apps

For one experiment, an online video was played through the platform of Youku, and power consumption was compared. The proposed mechanism was found to save 6.42% of the power consumption after the video playback finished. As seen in Fig. 11, power optimization mechanism achieved better performance for Wechat and Google chrome, because of a great deal of background data in such application. Specifically, the proposed mechanism reduced 13.64% energy consumption for the Wechat. In sum, our optimization mechanism performed better on background application than on foreground application.

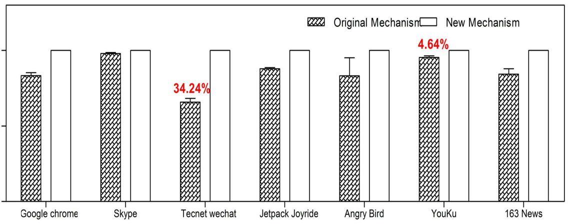

Figure 12. The comparison of transmission delay before and after use of optimization mechanism

In addition, the delayed transmission effect on latencies for packets was tested. The experiment was carried out by running App under different network environments. As shown in Fig. 12, the proposed mechanism consumed 4.64% of total time more than original mechanism for Youku. Specifically, it took 1.392 seconds more time compared with the original mechanism when playing the Angry Bird. More accurately, the proposed mechanism imposed greater influence on background packets (Chrome/ Wechat, etc.) than foreground packets (Skype/ Youku, etc.).

6. Conclusion

This paper analyzed the impacts of Wi-Fi RSSI on power consumption. The environment of wireless network has an impact on power consumption. The signal strength was classified as good, weak and bad. Under different signal strength conditions, the proposed mechanism aggregated packets and delayed transmission through prolonging the duration of smartphones stay in PS mode. Although the power consumption could be optimized by RSSI- awareness, there is no further distinction in task importance. For example, in the background, there are still some urgent tasks that need to be dealt with immediately (playing music /downloading file etc.). This could be optimized through the distinction between task behaviors more finely.

Acknowledgement. This work was supported by the Suzhou Scientific Research Program (No.SYG201731), and the National Natural Science Foundation of China (No.61772482, 61303206).

References

[1] Infographic.http://www.digitalbuzzblog.com/infographic-2013-mobile-growth-statistic.

[2] International Telecommunication Union, http://www.itu.int/ ITU-D/ict/statistics/ict/

[3] IEEE Standard for Information technologyÍCTelecommunications and information exchange between systemsÍCLocal and metropolitan area networksÍCSpecific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, IEEE, 2009, pp. c1 ÍC502.

[4] Ge Peng, Gang Zhou, All or None? The Dilemma of Handling WiFi Broadcast Traffic in Smartphone Suspend Mode. In Proceedings of the 34th IEEE Conference on Computer Communications, IEEE INFOCOM, Hong Kong, China, 2015, pp. 1212-1220.

[5] Ning ding. Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain. International Conference on Measurement and Modeling of Computer Systems , 2013, SIGMETRICS, pp. 29-40

[6] Roy Freedman, Alex Kogan, On Power and Throughput Tradeoffs of WiFi and Bluetooth in smartphones. IEEE Trans. on Mobile Computing, 2013, pp. 1363-1376

[7] Ashima Gupta, Prasant Mohapatra, Power Consumption and conservation in WiFi based Phones: A Measurement-Based study. 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 122-131

[8] Ekhiotz jon vergara, sanjuan. kernel level energy-efficient 3G Background traffic shaper for android smartphones. Wireless Communications and Mobile Computing Conference , 2013, IWCMC, pp. 443- 449

[9] Abhinav Pathak, Y. Charlie Hu, Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof. EuroSys, 2012, pp. 29-42

[10] Radhika Mittal, Aman Kansal, Empowering developers to estimate app energy consumption. Proceedings of the 18th annual international conference on Mobile computing and networking, Mobicom, 2012, pp. 317-328

[11] Andrei Gurtov. Effect of Delays on TCP Performance. IFIP Personal Wireless Communications, 2001. pp. 88-105

[12] Niranjan Balasubramanian, Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. ACM SIGCOMM, 2009, pp. 280-293

[13] Andrew J, Xin Qi, SAPSM: Smart adaptive 802.11 PSM for smartphones. UbiCom, 2012, pp. 11-20

[14] Bo Zhao, Wenjie Hu, Energy-Aware Web Browsing On Smartphone. IEEE Trans, on Parallel and Distributed Systems, 2015, 26(3), pp. 761-774

[15] Chulhong Min, Youngki Lee, PowerForecaster: Predicting Smartphone Power Impact of Continuous Sensing Application at Pre-installation Time. Sensys, 2015, pp. 31-44

[16] Lunde Chen, RSSI-Aware Energy Saving For Large File Downloading on Smartphone. IEEE embedded systems letters, 2015, pp. 63-68

[17] Pandaboard. http://pandaboard.org/content/pandaoard-es/

[18] WL1271 LS Research TiWi-BLE Transciever Module Datasheet. http://www.lsr.com/downloads/products/330-00 87.pdf, 2013

[19] T. Heinz. HIPAC: High Performance Packet Classification for Netfilter. 2004.