OPTIMIZING NFC PERFORMANCE ON A WEARABLE DEVICE USING EVOLUTIONARY ALGORITHM

A.V. Vaytulevich
Scientific Supervisor: Docent, Dr. A.V. Kudinov
Tomsk Polytechnic University, Russia, Tomsk, Lenin str., 2, 634050
E-mail: vaanval@gmail.com

ОПТИМИЗАЦИЯ РАБОТЫ МОДУЛЯ NFC НА НОСИМОМ УСТРОЙСТВЕ С ИСПОЛЬЗОВАНИЕМ ЭВОЛЮЦИОННОГО АЛГОРИТМА

А.В. Вайтулевич
Научный руководитель: доцент, к.т.н. А.В. Кудинов
Томский политехнический университет, Россия, г. Томск, пр. Ленина, 2, 634050
E-mail: vaanval@gmail.com

Аннотация. *Использование NFC в носимых устройствах может позволить использовать их в таких областях, как интерактивное обучение и системы безопасности. В настоящее время на рынке отсутствуют носимые устройства с полной поддержкой NFC. Включение NFC на таких устройствах вызывает повышенный расход батареи устройства. В данной статье предлагается метод настройки параметров NFC, учитывающий ограничения, накладываемый носимыми устройствами на тестирование конфигурации. В результате проделанной работы была создана конфигурация NFC, близкая к оптимальной.*

Introduction. The last few years have seen a considerable growth in the amount of consumer wearable devices equipped with various sensors. By obtaining a functioning NFC module on such device, it would become possible to use them in contactless payment systems or facilitating access control by replacing smart cards [1]. Despite benefits from these applications there are currently no devices that both have a programmable NFC chip on board and provide any kind of API to control it [2]. One of such systems is the recent Sony SmartWatch 3 running on Android. The whitepaper released by its manufacturer suggests that its system on chip (SoC) may have a fully functional NFC controller built into it [3]. During one of our projects, we have successfully used Sony SmartWatch 3 as an NFC-reading device. In a collective effort between Gazprom transgaz Tomsk, Ltd. and Tomsk Polytechnic University, we have created a system that was designed to assist teaching students operating a gas distributing station. This was achieved by installing customized software that is part of the NFC software stack used by most NFC-supporting Android smartphones [5]. The major problem reported by users was severely increased battery consumption; therefore, a method of configuring the device to facilitate these problems had to be proposed.

Research area analysis. We began to implement our project using Sony SmartWatch 3 as a wearable NFC sensor, since it was the only one claiming to have NFC built into it. Sometime after the initial release Michael Roland proposed that the watch may contain a programmable NFC module in [6] which implies that it can be accessed by custom application and used to exchange data with NFC tags and other NFC-enabled devices. The stock operating system, however, lacked most of supporting software, which meant that NFC could not be used by our application. To address this, we have used an NFC support stack provided for most Android devices with required hardware [4].

There are approximately 60 parameters defining the work of hardware NFC module within a single configuration file [5]. Some of them can potentially affect overall NFC performance (including effect on device battery life); however, there are no direct recommendations on their effect on the indicators that have to be improved. This suggests creating a system that can conduct automatic tuning of NFC configuration. Therefore, the goal of this research is to design and implement an algorithm that is capable to automatically create and test various NFC configurations on a real Sony SmartWatch 3, while reducing overall battery life impact.

Experimental setup. A straightforward way to approach this is to decrease polling rate. A series of tests were conducted to estimate possible effect of this. Despite this being effective, the effect turned out to be insufficient compared to a stock device without NFC (see table 1). The battery life was reduced from 95.1 % to 64.6 % which still is considerably less than that of a device with stock NFC support. Additionally, further decreasing polling rate turned out to be impractical since responsiveness (i.e. average time required to read a tag after the watch has been put close to it) decreased to more than 2 seconds.

Table 1

The Effect of modifying Polling Rate on Battery Life for Sony SmartWatch 3

Polling rate, Hz	Battery life decrease, %	Average read time, ms
2	94,3	684,2
1,33	78,6	891,9
1	64,6	1260,4

The battery life B and average response time R were used as a fitness estimation of a configuration being tested; the configuration fitness F is defined in (1).

$$F = B + R / 2000 \qquad (1)$$

The B parameter in (1) is normalized, whereas R must additionally be normalized over the maximum range of 2000 milliseconds (determined from maximum response time for reading an NFC tag). Finding an optimal configuration for NFC would then mean finding a configuration with minimal fitness F. Failing to read at least one tag (or having the NFC service stop unexpectedly) would result in marking the candidate as failing (i.e. setting its F to a value high enough to block it from further usage).

Implementation. Of the total of 60 NFC parameters, 20 parameters were left; out of which 8 were continuous, 5 were discrete and 7 were vectors of discrete values. The average number of possible values was determined to be 6 (rounded to nearest integer) for discrete parameters, 10 for vectors of discrete parameters. Therefore, rough search space estimation for discrete values only is $2 \cdot 10^{11}$. With minimum time for testing a single candidate configuration of approximately 3 minutes, the time required for exhaustive search would be deniably high.

To address this issue, an algorithm capable of combing best parameters from best candidate configurations was required. We decided to use a modified genetic algorithm; by doing so, we would allow defining a way for candidate solutions to inherit best parameters from previous iterations by means of crossover and mutation. For this algorithm, configurations were considered individuals, their genomes consisting of a set of parameters that need to be optimized [6]. Crossover was defined as fully exchanging genes between two parent individuals (since rules mixing of each individual parameter would have to be defined otherwise).

To increase algorithm convergence we suggest a way to prevent parameter degrading by using a tree containing the history of all tested parameter values. Each change in configuration is assigned an estimate of its individual effect on overall configuration fitness. This estimation is calculated using (2), where P_{ij} is the resulting probability that parameter i has contributed into achieving fitness j and N_j is the number of parameters that were changed compared to both of parent individuals.

$$P_{ij} = 1 / N_j , \quad (2)$$

This probability was then used to modify the probability of a given parameter value to be promoted during crossover (see (3), where P_{cross} is the probability of successful parameter crossover and P is overall crossover probability). By using this approach we penalize unfit parameters from propagating to further generations.

$$P_{cross} = P \cdot P_{ij} , \quad (3)$$

The system was run and completed work by reaching configuration fitness plateau in 65,93 hours. A total of 3 recharges were required to complete this work. A total of 1168 candidate configurations were tested over the run spanning across 41 generations. The resulting battery life decrease was reduced to 19,1 % from the initial 95,1 % (see table 2).

Table 2

Comparing Goal Indicators for Optimized Configuration

Device	Average life on battery, s	Responsiveness, ms	Battery life decrease, %
Sony Xperia Z1	108144,16	n/a	9,2
SW3, Default	4511,67	684,2	95,1
SW3, Optimized	74921,00	2521,9	19,1

Results. The result of this work is an approach to automated configuring of NFC module parameters on a device with limited capabilities of model fitness assessment. A way of reducing the amount of required tests was proposed by means of increasing algorithm convergence basing on penalizing genes that drastically reduce model fitness and promoting genes that do otherwise. The proposed algorithm was used to generate a near-optimal configuration for NFC-enabled Sony SmartWatch 3, showing improved performance in all three of conducted tests. The project goals were achieved, however, the device never showed performance improvements comparable to smartphones with NFC support.

REFERENCES

1. Kim H.J., Park J.H., Lee J.Y., Ryou J.C. Biometric Authentication Technology Trends in Smart Device Environment // Mobile and Wireless Technology. – 2015. – P. 199–206.
2. Heath A., Apple Watch will replace your car keys, says Tim Cook [Electronic resource] // The Telegraph. – Retrieved 14.04.2015 from http://www.telegraph.co.uk/technology/apple/watch/11439847/Apple-Watch-will-replace-your-car-keys-says-Tim-Cook.html.
3. Sony Xperia – Official website: Whitepaper Sony SmartWatch 3 [Electronic resource]. – Retrieved 11.11.2015 from http://www-support-downloads.sonymobile.com/swr50/whitepaper_EN_swr50_smartwatch3_2.pdf.
4. Roland M. Is there NFC on Sony SmartWatch 3 (SWR50)? [Electronic resource] // Josef Ressel Center for User-friendly Secure Mobile Environments. – Retrieved 28.01.2016 from https://usmile.at/blog/is-there-nfc-on-sony-smartwatch-3-swr50.
5. NXP Semiconductors – Linux Software Stack Integration Guidelines [Electronic resource]. – Retrieved 28.01.2016 from http://www.nxp.com/documents/application_note/AN11697.pdf.
6. Dieng M., Azais F., Comte M. et al. Study of Adaptive Tuning Strategies for Near Field Communication (NFC) Transmitter Module // Proceedings of the 19th International Mixed-Signals, Sensors and Systems Test Workshop (IMS3TW). – 2014. – P. 1–6.