

Bandwidth Research of Wireless IoT Switches

Zhengbing Hu

*School of Education Information Technology
Central China Normal University
Wuhan, China
hzb@mail.ccnu.edu.cn*

Volodymyr Buriachok

*Department of Information and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
v.buriachok@kubg.edu.ua*

Volodymyr Sokolov

*Department of Information and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
v.sokolov@kubg.edu.ua*

Bohdan Vovkotrub

*Department of Information and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
bvovkotrub@gmail.com*

Yevhen Zotkin

*Department of Information and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
evgeniy.zotkin199702@gmail.com*

Abstract—The paper presents the research and comparative analysis of the bandwidth of low-power wireless IoT devices as wireless switches. Investigated and analyzed sensors the dependence of FTP multimedia data transmission speed on wireless Wi-Fi network on the temperature of the switch processor. To get temperature indicators sufficiently built into Python libraries to read temperature files. The paper focuses on the synchronicity of measurement results records for more accurate analysis. As a result, the dependence of the measured factors was calculated using the Pearson correlation formula. These measurement factors affect the autonomy and energy consumption, which is important for IoT devices, and therefore, among the devices tested, recommendations were made regarding their choice when used depending on the conditions.

Keywords—*wireless network; bandwidth; network load; IoT; Raspberry Pi; RPi.*

I. INTRODUCTION

To date, widespread IoT devices are capable of automating various processes without human involvement. Such systems typically use low-power microcomputers and, along with them, wireless networking technologies. Reliability and speed of delivery of information is an important problem when it is transmitted in such systems with the lowest energy consumption. One of the most widespread microcomputers is the Raspberry Pi (RPi) [23]. They are used as a smart home, connecting different sensors, in data centers as a measurement of temperature readings in server rooms, CCTV and more. Therefore, RPi 3 Model B [1] and RPi Zero W [2] are used as the study.

II. RELATED WORKS

Consumption of current and voltage is an important issue, as it depends on the length of time provided that the device is powered by a battery, such as in [3], which exchanged data between the user and the access point on a wireless network. If you use this example to exchange large volumes of data, then the question arises about their fast delivery. Also, increasing the amount of data will increase the load on the switch, which means that the power consumption will be increased. Therefore, you need to choose a device that will consume less power and thus have the maximum capacity [24].

Another problem with wireless data transmission is the distance between the client and the access point [22]. After all, as the distance increases, the number of interference increases, which will reduce the speed and reliability of data transmission. In [4] is given an example of what is better to use shielded devices for information. Thermal conductivity played a significant role in this work, as its failure led to one of the microcontrollers being burned [21]. Therefore, a low-power device should be more durable to increase the temperature that affects the throughput, which will be explored in this paper.

III. DEVELOPMENT OF EXPERIMENTAL MODEL

For the sake of pure experiment, both machines are configured identically to the same operating system and the same settings. Therefore, all that is written as RPi (this refers to RPi Zero and RPi 3). In order to have a minimum system load, the OS was used as an operating system without a graphical interface—Raspbian OS Lite. The peculiarity of this OS is the lack of “unnecessary” programs that slow down your

computer. In this version of OS there is a package for work with GPIO. RPi act as a wireless Wi-Fi hotspot, configured as in [5]. The access points have an vsftpd FTP server that is easy to set up, fast and secure. Deployed on a separate machine, the FTP client downloaded the same file from RPi at different distances: 0.5 m and 5.0 m, 10 measurements at each distance. As a result, the average of all tests will be taken into account. Every five of the metrics are taken off: server CPU load and temperature, server current and voltage consumption, and file upload speed on the FTP client (Fig. 1).

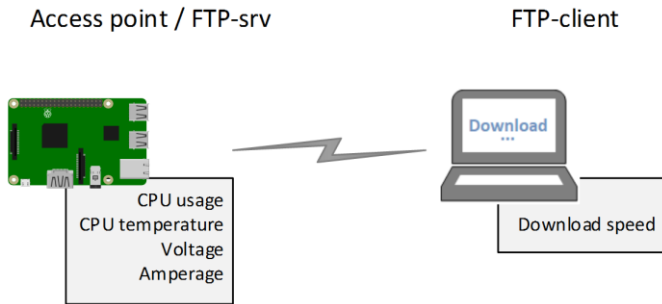


Fig. 1. Scheme of the experiment.

The second step was to automate the process of recording history files. This will minimize human error when recording the results, and thus improve measurement accuracy and recording speed. The INA219 digital sensor was used to measure voltage and current. In addition to the built-in processor temperature sensor, an external DS18B20 sensor is additionally used. Both sensors are connected to the GPIO interface, which allows to receive information in the text format used for measurement automation (Fig. 2). Initially, it was planned to measure voltage and current using a USB tester, but the problem is that they cannot be recorded in electronic text format. There were two options for solving this problem: the first one to take a photo of the tester sensors every 5 seconds and then process these images, and the second option to use sensors with a GPIO interface. The first option is more resource-intensive, since it takes more time and processing power to process images than when receiving information through GPIO RPi interfaces. Therefore, the second option was chosen for the experiment.

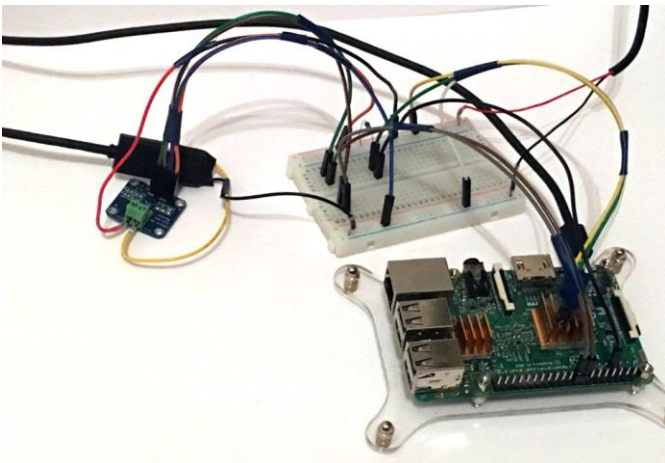


Fig. 2. General view of the test equipment.

In order for the results to be accurate on both machines, they must be started at the same time and run synchronously and recorded at regular intervals. The problem was solved using the software interface of synchronous communication between processes (TCP sockets). The processes of such exchange can be performed on different computers connected by a network.

Next, we will discuss the process of developing automation using the *sensors.py* scripts on the server side, which removes the sensors from time intervals of 5 s, and *ftp_clt.py* on the client side, which downloads the file from the FTP server and logs the download speed from at the same intervals. These machine scripts interact with a TCP socket using the following algorithm:

1. The server listens to port 5555.
2. The client sends a "Start" command to this port.
3. The server receives a command and both start measuring at the same time.

IV. PROTOTYPE DESIGN

The *sensors.py* script is running on Server-PC, which receives data from DS18B20 [10] and INA219 [11] sensors. The *ina219* library is used to work with INA219. The *psutil* library is used to get the CPU utilization status. When the DS18B20 temperature sensor is connected, the `/sys/bus/w1/devices/<serial_number_of_DS18B20>/w1_slave` file is automatically created to display the temperature in text format. Therefore, in order to get ambient temperature in *sensors.py*, it is enough to open and parse this file using the built-in OS library in Python. In addition, when the *vcgencmd measure_temp* command is initialized, the processor temperature is also output from the built-in sensor using OS. The *get_sensors()* function uses the above functions, and every 5 s displays information from the sensors and logs it. It is initialized after the socket library receives a message from the client: "FTP is running." The flowchart of *get_sensors()* is shown in Fig. 3.

When you start *ftp_clt.py*, you first connect to the FTP server, then searches for the file you want to download. In order to monitor the process of downloading the file, you need to get the full file size on the server using *ftp.size(<file>)* is the file size is compared on the client and the server. After connecting to the FTP server, *ftp_clt.py* sends a message to the server: "FTP is running" via socket to the server. The *sensors.py* script receives this message and, together with *ftp_clt.py*, shuts off TCP sockets and starts measuring readings at 5 s interrupts. In order for *ftp_clt.py* to load a file at the same time using the *ftp_download(<file>)* function and to measure the file load speed in parallel with *speedtester(<file_size_ftp>, <file>, <sleeping>)*, the *process* module from the multiprocessing library is used. All measurement results are displayed on the screen and logged.

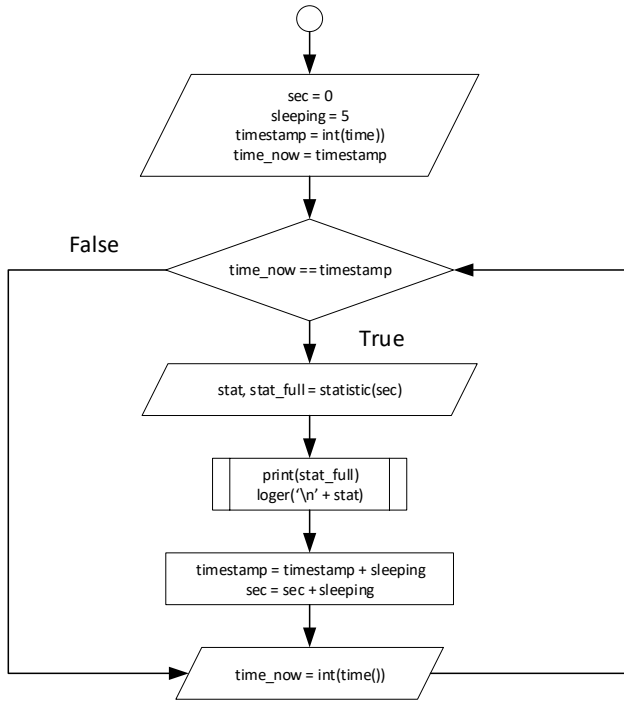


Fig. 3. Flowchart of the getting sensor function.

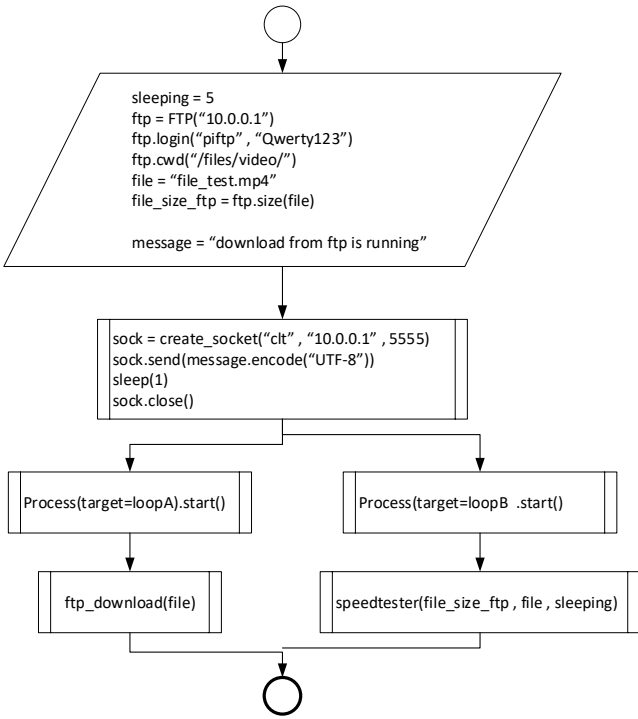


Fig. 4. Flowchart of the main script.

V. MEASUREMENT RESULTS

At a distance of 0.5 m from the access point, the 655 MB file download took place at approximately the same time on both devices, but the RPi 3 schedule was more straightforward than at 5.0 m (Fig. 5).

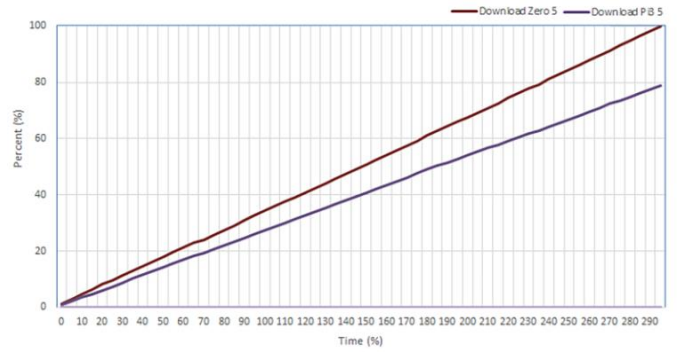


Fig. 5. Comparison of download status over time at 5.0 m.

The processor on RPi 3 was less downloaded than on RPi Zero. An interesting fact is that at a longer distance of the client from the access point, the RPi 3 processor load was less, and the RPi Zero at a longer distance was periodically greatly reduced but aspired to the maximum value (Fig. 6). Consider how the temperature of the processors changed. At times when the load was decreasing, the temperature was also falling—this may mean that the trotting—a mechanism to protect the processor from overheating (with the omission of cycles), worked. On RPi 3 the CPU load is smoother.

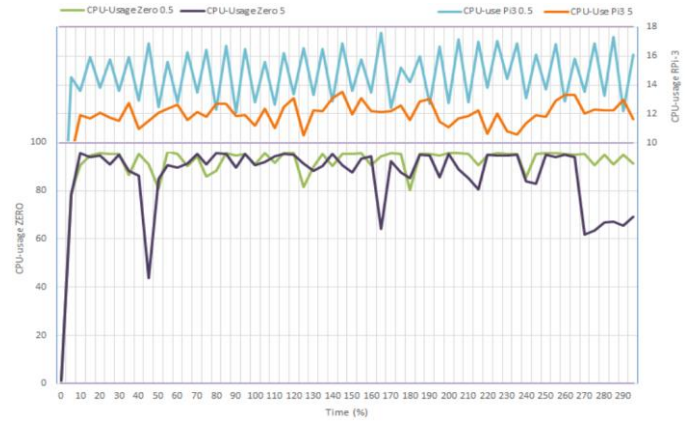


Fig. 6. Load comparison of RPi Zero and RPi 3 CPUs.

The most noticeably decreased injection speed with a sharp decrease in current consumption. The current consumption increased sharply when it was necessary to reduce the processor load to reduce the processor temperature (Fig. 7).

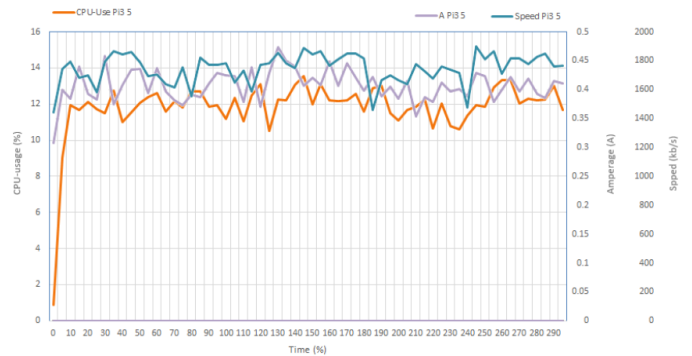


Fig. 7. CPU load, power consumption, and RPi 3 file download speed at 5 m.

VI. IMPLEMENTATION

Using the Pearson correlation, you can calculate the dependence of factors among themselves:

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}} \quad (1)$$

where x is the value of one factor; y is the value of another factor to which the ratio applies.

In the Table I shows the results of calculating the Pearson correlation coefficient for different distances by (1). The correlation coefficient varies in the range from plus one to minus one. If there is a positive correlation, an increase in one indicator increases the second. With a negative correlation, an increase in one indicator entails a decrease in another. The larger the correlation coefficient module, the more noticeable the change of one indicator reflects the change of the second.

TABLE I. THE PEARSON CORRELATION COEFFICIENT FOR IOT SWITCHES

0.5 m distance for RPi Zero / RPi 3					
	Speed	CPU usage	CPU temp.	Ext. temp.	Voltage
CPU usage	0.12/0.05	—	—	—	—
CPU temp.	0.31/0.24	0.53/0.33	—	—	—
Ext. temp.	0.47/-0.21	0.24/0.16	0.84/-0.03	—	—
Voltage	-0.03/-0.19	0.02/-0.36	-0.11/0.04	0.03/0.01	—
Amperage	-0.13/0.01	0.15/0.39	0.24/0.05	0.12/0.27	-0.03/-0.34
5.0 m distance for RPi Zero / RPi 3					
	Speed	CPU usage	CPU temp.	Ext. temp.	Voltage
CPU usage	0.55/0.38	—	—	—	—
CPU temp.	0.30/0.24	0.26/0.46	—	—	—
Ext. temp.	0.11/0.02	-0.02/0.05	0.87/0.19	—	—
Voltage	-0.16/-0.04	0.06/-0.17	-0.18/-0.08	-0.18/-0.04	—
Amperage	0.05/0.32	0.01/0.43	0.22/0.17	0.21/-0.07	-0.12/-0.10

VII. IMPLEMENTATION

The work done does not exhaust the full depth of the study. Possible areas for further research include a more sophisticated approach to information retrieval and a deeper statistical analysis of the data obtained. This technical complex in the form of RPi together with the temperature sensor DS18B20 and the current, voltage and power sensor INA219 can be integrated into such systems as monitoring systems in data centers, "smart home" systems, etc. This technical complex is already used for temperature monitoring server in the data center. As an example, the Grafana platform shown in Fig. 8 is used to visualize, monitor and analyze the data obtained.



Fig. 8. Grafana dashboard with temperature and voltage indicators.

VIII. CONCLUSION AND FUTURE WORK

Based on the results obtained, it can be concluded that all the factors studied, namely server CPU load, processor temperature, current consumption, affect the file download speed. When the processor is running at peak temperature, there is a process of overheating protection, called clocking. At this time, the processor clock speed decreases and its performance and efficiency are reduced. This results in a slower download speed of the file. The data shows that in RPi Zero W the process of clock throttling occurred with a higher decrease in clock speed and processor performance than in RPi 3. For a large amount of wireless data transfer, the RPi 3 is more suited than the RPi Zero W because the third version is more productive, so it can support more concurrent tasks. However, for simple tasks, RPi Zero is more suitable because this version requires less power. For IoT devices, autonomy and power consumption is an important indicator.

ACKNOWLEDGMENT

This scientific work was partially supported by RAMECS and self-determined re-search funds of CCNU from the colleges' primary research and operation of MOE (CCNU19TS022).

REFERENCES

- [1] A. k and U. B. Mahadevaswamy, "Automatic IoT Based Plant Monitoring and Watering System using Raspberry Pi," *International Journal of Engineering and Manufacturing*, vol. 8, no. 6, pp. 55-67, Nov. 2018. doi: 10.5815/ijem.2018.06.05.
- [2] Raspberry Pi Foundation. (2015, Apr.). "Raspberry Pi 3 Model B." [Online]. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> [Nov. 27, 2019].
- [3] Raspberry Pi Foundation. (2017, Feb.). "Raspberry Pi Zero W." [Online]. <https://www.raspberrypi.org/products/raspberry-pi-zero-w/> [Nov. 27, 2019].
- [4] V. Yu. Sokolov and D. M. Kurbanmuradov, "Methods of Counteracting Social Engineering at Objects of Information Activity [Metodyka protydyiyi sotsial'nomu inzhynirynhu na ob'yektakh informatsiyonoyi diyal'nosti]," *Cybersecurity: Education, Science, Technology*, no. 1, 2018, pp. 6-16. doi: 10.28925/2663-4023.2018.1.616.
- [5] J. Kaur and K. Kaur, "Internet of Things: A Review on Technologies, Architecture, Challenges, Applications, Future Trends," *International Journal of Computer Network and Information Security*, vol. 9, no. 4, pp. 57-70, Apr. 2017. doi: 10.5815/ijenis.2017.04.07.

- [6] D. Johnson and M. Ketel, "IoT: Application Protocols and Security," *International Journal of Computer Network and Information Security*, vol. 11, no. 4, pp. 1–8, Apr. 2019. doi: 10.5815/ijcnis.2019.04.01.
- [7] V. Yu. Sokolov, "Comparison of Possible Approaches for the Development of Low-Cost Spectrum Analyzers for 2.4-2.5 GHz Sensor Networks [Porivnyannya mozhyvykh pidkhodiv shchodo rozrobky nyz'kobyudzhetnykh analizatoriv spektru dlya sensorykh merezh diapazonu 2,4–2,5 HHts]," *Cybersecurity: Education, Science, Technology*, no. 2, 2018, pp. 31–46. doi: 10.28925/2663-4023.2018.2.3146.
- [8] S. Revathi, "Protocols for Secure Internet of Things," *International Journal of Education and Management Engineering*, vol. 7, no. 2, pp. 20–29, Mar. 2017. doi: 10.5815/ijeme.2017.02.03
- [9] Oestoidea. (2017, Sept.). "Access Point on Raspberry Pi 3 with Parameter Display." [Online]. Available: https://github.com/Oestoidea/Adafruit_Python_SSD1306 [Nov. 27, 2019].
- [10] Les Pounder. (2017, Jun.). "DS18B20 Temperature Sensor With Python (Raspberry Pi)." [Online]. <https://bigl.es/ds18b20-temperature-sensor-with-python-raspberry-pi/> [Nov. 27, 2019].
- [11] Python Software Foundation. (2018, Jun.). "pi-ina219 1.2.0. Project description." [Online]. <https://pypi.org/project/pi-ina219/> [Nov. 27, 2019].