Investigation of Availability of Wireless Access Points based on Embedded Systems

Feodosiy Kipchuk, Volodymyr Sokolov,
Volodymyr Buriachok, Pavlo Skladannyi
Dept. of Inform. and Cyber Security
Borys Grinchenko Kyiv University
Kyiv, Ukraine
ORCID: 0000-0003-4816-9246
ORCID: 0000-0002-9349-7946
ORCID: 0000-0002-4055-1494
ORCID: 0000-0002-7775-6039

Lidia Kuzmenko
Center for Long-term Plan. and Mon. of Educ. Activities
National Academy of the Security Service
Kyiv, Ukraine
ORCID: 0000-0001-7392-0324

***Abstract*—The paper presents the results of load testing of embedded hardware platforms for Internet of Things solutions. Analyzed the available hardware. The operating systems from different manufacturers were consolidated into a single classification, and for the two most popular, load testing was performed by an external and internal wireless network adapter. Developed its own software solution based on the Python programming language. The number of wireless subscribers ranged from 7 to 14. Experimental results will be useful in deploying wireless infrastructure for small commercial and scientific wireless networks.**

Keywords—embedded system; threat; wireless network; vulnerability; security; operating system; service.

I. Introduction

The purpose of this work is to solve the problem of “last mile.” Taking into account the specifics of wireless device-based network devices, we emphasize the importance of ensuring a stable connection of devices, protecting their work from unauthorized access and preventing and minimizing interference to normal functioning. Using minicomputers allows you to save a lot of resources. Such systems require a minimum of power, network resources, flexible setup. It also has a very wide functional extension, through additional modules. However, embedded systems have certain limitations, such highly loaded systems as server parts and services will work on a small scale. However, with the logical construction of a functional network from such devices, as clients, agents, or sensors, mini computers can be combined into larger and scalable networks [1].

Currently, a large number of devices are capable of interacting of each other, which makes them multifunctional. Such devices are capable of operating independently, in groups, acting as an element of a particular network, also known as Internet of Things (IoT).

The paper is structured as follows. Section II and III gives an overview of related and future work. In section IV, has presented our vision of the hardware platform choosing for which available operating systems and hardware for embedded systems were analyzed. Section V gives an overview of the software and presents a software architecture. In section VI, has described hardware and software implementation of the system. Sections VII and VIII present results of the comparative analysis of frequency ranges and operating systems. This paper ends with section IX where concluded current state of development of this system and described directions of future development.

II. Related Works

A. General Overview

Returning to the key aspects of computer systems, you need to consolidate the following knowledge and determine: how the embedded systems work, which tasks will solve your device, architecture and operating principle, the need for additional modules, uninterrupted work, an adjustment of the required level of protection and vulnerability prevention, project development potential and scalability.

B. Specific Practical Examples

Wireless devices may be vulnerable to poor coverage of the network. This disadvantage allows at least to build three attack vectors with the fake access point, user substitution, end-device spoofing or access point, network clogging, and others. This work is a good example of security measures and prevention of such attacks [2]. The solution provides a tool for scanning and detecting counterfeit access points and provides the necessary user access permission to block access to user files ad hoc on the Wi-Fi Data Link Layer. Since the writing of this article, more than five years have passed. Modern hacking tools have been greatly improved during this time, so this solution requires new testing for stability.

This topic also covers the other vulnerabilities of wireless networks that have the same intention—denial of service DoS or DDoS [3]. This architecture is not new at this time. Currently, a new solution and a more solid architecture are needed to help prevent DoS attacks. Despite the current releases of upgraded 802.11 wireless security standards, the devices still have many vulnerabilities, so it’s time to conduct analyzes and experiments on many known types of DoS attacks. For this purpose, an algorithm called the Alternative Numbering Mechanism (ANM), which prevents DoS attacks.

For more tuned networks also need to control data flow routing and managing network-level VLAN [4]. The solution is positioned as a new strategy for security

978-x-xxxx-xxxx-x/19/$31.00 ©2019 IEEE

measures. This is provided by another encryption algorithm for unshared key and virtual local area network.

By applying all possible security measures, you might be sure that the chances of hacking the network are minimized and matched to a certain level. It is advisable to anticipate impedance of ISO standards in advance. Taking into account the above requirements this paper will consider implementation, for which the built-in systems were assigned. Microcomputers are used in the daily tasks of enterprises, have taken their place of honor with their flexibility, and are applied even in space, called “Astro PI Mission Zero code,” which operates at the International Space Station. A good example in the industry is the use of microcomputers in a combination of servo motors, sensors, controllers in poultry farming [5]. Such small systems can provide performance even in the most demanding conditions. Raspberry Pi (RPi), together with a humidity sensor, servo controller and light bulbs, can carry out work as an artificial incubator. In addition, the flexibility of the operating system (OS) setting allows you to download or develop a program to monitor humidity and temperature and synchronize data through the cloud service on Android.

III. Previous Work

One of the good examples is the use of the platform as a network access point with the use of a mini screen, which can display the necessary monitoring functions. The access point based on the *Raspbian OS* requires little resources represented in the work. Using the on-screen module and programming allows you to display almost any information, for example: connected devices, gateway connection status, device popup addresses, IP addresses of devices, etc. [6].

IV. Hardware Platform Choosing

The RPi computers were a pioneer in manufacturing microcomputers, and it has approved the standard and is still the most widely used devices for many projects and has the largest community support, most different OS supported (Fig. 1). This is main reasons, why we choose a latest version of RPi 3 B+. There is another board, for instance, Banana Pi, Arduino, ODROID, ESP32, etc. Comparative analysis of boards is presented in Table I.

Because of familiarization with the characteristics of the hardware from the manufacturer documentation, RPi was chosen for the experiments.

V. Software Architecture

The software should provide:

- Check the possibility of simultaneous loading of several clients, with a stable load.
- Check the maximum bandwidth of the system as an access point.
- Check the quality of service at full load.
- Collect statistics for calculating formulas and coefficients.
- Testing Python code in limited resources.

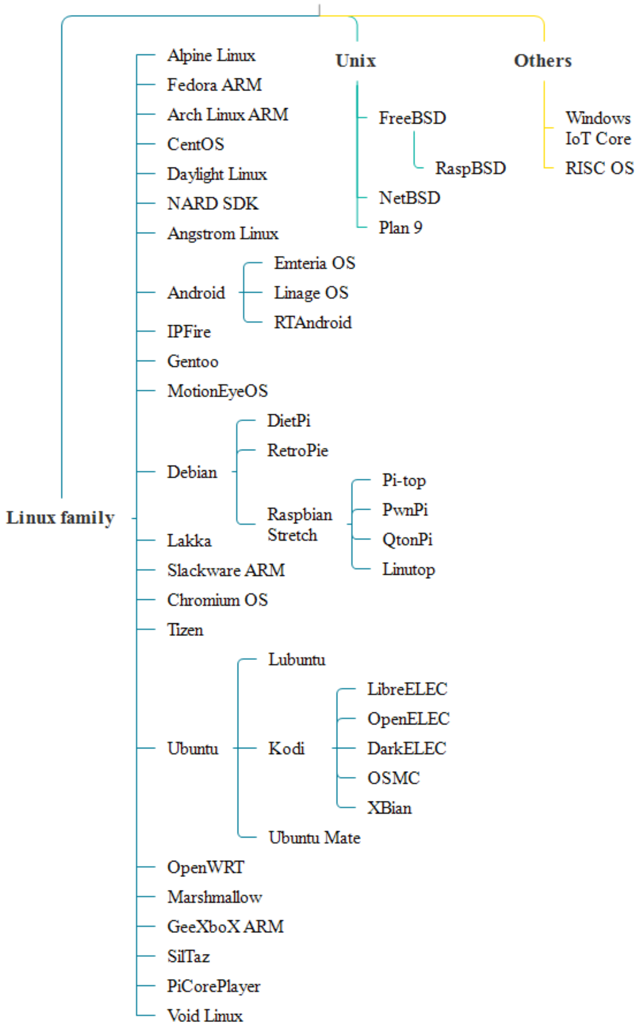

Fig. 1. Operating system classification.

A very large number of operating systems already existed, and the choice was too large. The choice was towards the maximum stable and adapted operation of drivers.

VI. Hardware and Software Implementation

As an external network card, the TP-Link TL-WN722N with the AR9271 chip specs file with an external antenna was working [7]. In practice, this network card has been limited enough to achieve the goals of the experiment. As a result, the platform could not work simultaneously with more than 8 devices, 1 server, and 7 clients. But the built-in CYW43455 controller supports twice as many as 14 clients and 1 server.

A very large number of operating systems already existed, and the choice was too large. The choice was towards the maximum stable and adapted operation of drivers.

TABLE I. General Parameters of IoT Boards

Board	Processor	Freq, GHz	RAM, Gb	Eth, Mb/s	Band	Price, USD
RPi 3B+	BCM2837	4×1.40	1.0	0.1	dual[b]	45
RPi Zero	BCM2835	1×1.00	0.5	0.1[a]	single	25

Board	Processor	Freq, GHz	RAM, Gb	Eth, Mb/s	Band	Price, USD
Omega2 Plus	MIPS	1×0.58	0.1	1.0[a]	single	60
Rock64 Media Board	RK3328	4×1.50	4.0	1.0	dual[b]	60
Arduino MKR 1010	SAMD21	1×0.05	0.2	0.1[a]	single	50
Le Potato	Amlogic S905X	4×1.50	2.0	0.1	dual[b]	60
BBC micro:bit	Cortex-M0	1×0.02	0.2	—	single	25
Pine A64-LTS	RK3399	6×1.20	4.0	1.0	dual[b]	80
Banana Pi M64	Allwinner A64	4×1.20	2.0	1.0	single	65
Odroid-C2	Amlogic S905	4×1.50	2.0	1.0	single[b]	70
Orange Pi Plus2	Allwinner H3	4×1.60	2.0	1.0	single	60
Rock PI 4 B	Rockchip RK3328	4×1.80	4.0	1.0	dual	75
NanoPC-T3 Plus	S5P6818	8×1.40	2.0	1.0	single	100
Odroid-XU4	Exynos 5422	4×2.00	2.0	1.0	dual[b]	85
Tinker Board S	Rockchip RK3288	4×1.80	2.0	1.0	single	90
Latte-Panda	Cherry Trail Z8350	4×1.80	2.0	1.0	single	100
Minnow-board Turbot	Atom E3826	2×1.50	2.0	2×1.0	dual[b]	230
Beagle-Board-X15	AM5728	2×1.50	4.0	2×1.0	dual[b]	240
Huawei HiKey 960	Kirin 960	8×2.30	3.0	1.0[a]	dual	300

[a] available with external Ethernet module

[b] available with external Wi-Fi module

As an external network card, the TP-Link TL-WN722N with the AR9271 chip specs file with an external antenna was working [7]. In practice, this network card has been limited enough to achieve the goals of the experiment.

As a result, the platform could not work simultaneously with more than 8 devices, 1 server, and 7 clients. But the built-in CYW43455 controller supports twice as many as 14 clients and 1 server.

The software operation algorithm in Business Process Model and Notation (version 2) is presented in Fig. 2.

We used the following software in the experiment:

- RPi 3B+ (2.4/5 GHz) with *Raspbian Lite OS* or *OpenWRT*.
- MicroSD 16 Gb UHS-I as a storage.
- TP-Link TL-WN722N v. 1.
- Power bank.

The following software tools were identified to implement the access point:

- *hostapd* is service for Wi-Fi access point.
- *dnsmasq* is DHCP-server and DNS.
- *ath9k-htc* is driver for TL-WR722N.
- *vsftpd* is FTP-server.

At the beginning of the *OpenWRT* experiment, the version 18.0 was released, but it did not work correctly: when setting up an access point, the gateway configuration was not stored, so was taken the last stable version 17.0. But a month later, errors were fixed, and version 18.01 was successfully installed and worked correctly (Fig. 3).

To simulate a stable load close to the working conditions in the office of the company, the method of generating traffic to the client-server connection using FTP was selected: clients download one large file at a time (because using file sizes of different sizes may result in failures at transmission rates and errors).

When selecting an FTP-server, several programs were checked, and the choice stopped at *vsftpd*, which is easy to set up and does not have a customer database and additional settings. Also, *vsftpd* uses a minimal amount of operating resources and is fairly safe, efficient, and does

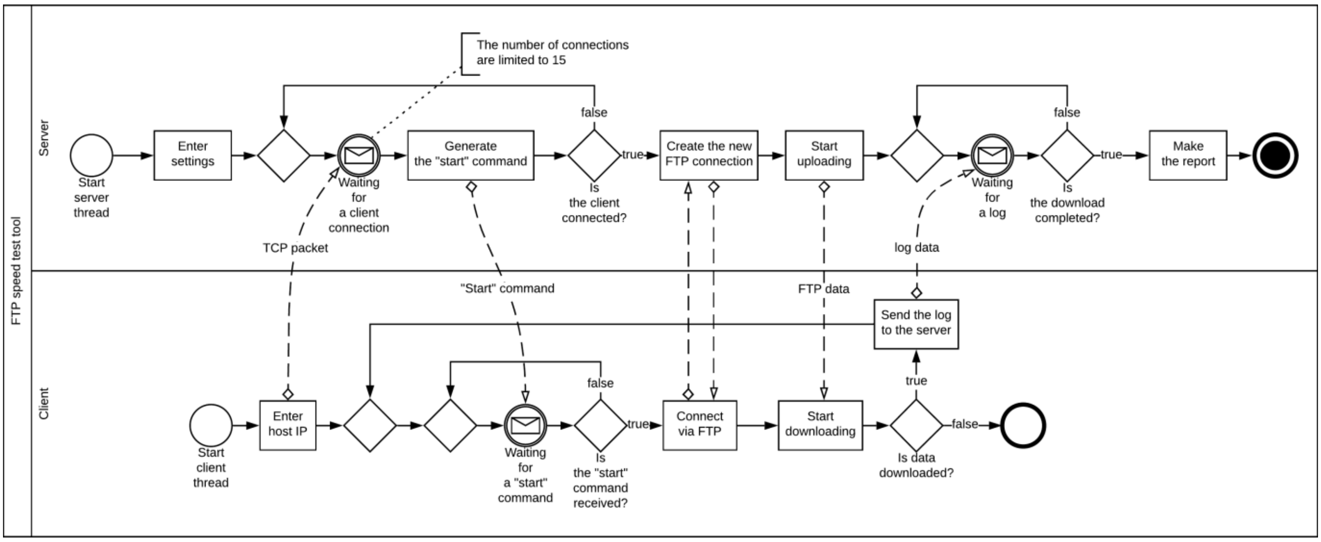

Fig. 2. Software functional scheme.

not require additional services unlike *ProFTPD*.

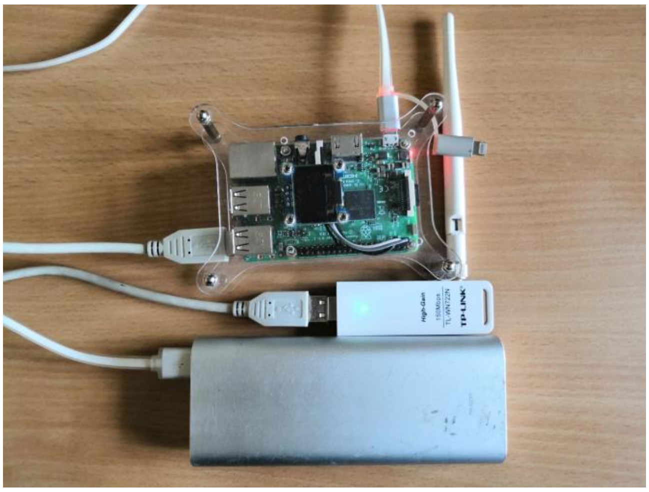

Fig. 3. Experimental hardware.

A file size of 30 MB has been selected, for complete download tracking for all clients. Also, theoretically, this size allows the channel to be used by all users at a time, at least 90 seconds with 7 connected devices.

In addition, we can compare the diagram of speed tests on different wireless cards and OS's in Fig. 4.

The test environment was a closed room, in which 15 PC's was located. In the middle of the room, there was an access point and one of the PCs that served as the server. The maximum distance from the access point to the distant PC reached 6 meters, which is an approximate value to work in the office space.

At the time of testing, all unplanted programs were disconnected from all stations and network activity was minimized.

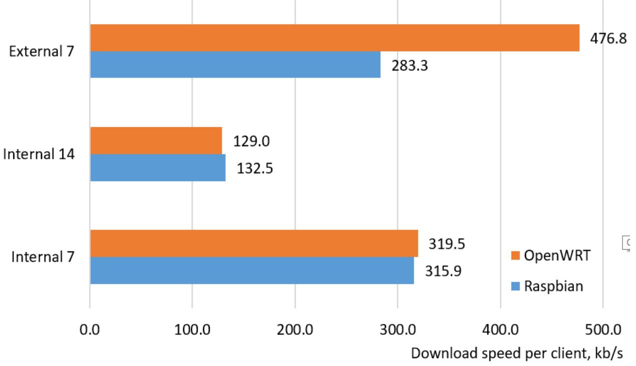

Fig. 4. Speed test with different operating systems and cards.

All PCs are Dell OptiPlex 3050 Micro with OS Windows 10 Education, controller Intel® Dual Band Wireless AC 3165 (802.11ac) 1×1 with external antennas.

VII. Comparative Analysis of Frequency Ranges

Having built a polynomial trend line (see Fig. 5), it is easy to see that the relative decrease in the transmission rate for both bands is almost the same and is about three times (the absolute speed drop for 2.4 GHz frequency band was 0.04 MB/s, and for 5 GHz was 0.18 MB/s).

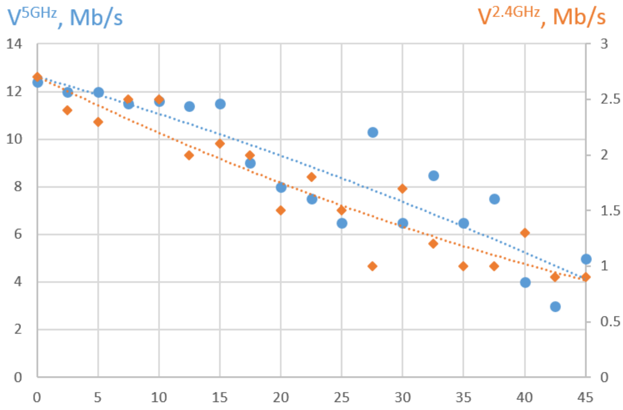

Fig. 5. Data transmission test on different frequencies.

The graph of comparison of the controller's operating at different frequencies was conducted in a closed room, with a total length of up to 50 m, indirect visibility without interference. In a partially charged range of 2.4 GHz and a free range of 5 GHz. Channel 11 has been selected for 2.4 GHz, and 5 GHz channel 36 (Fig. 6).

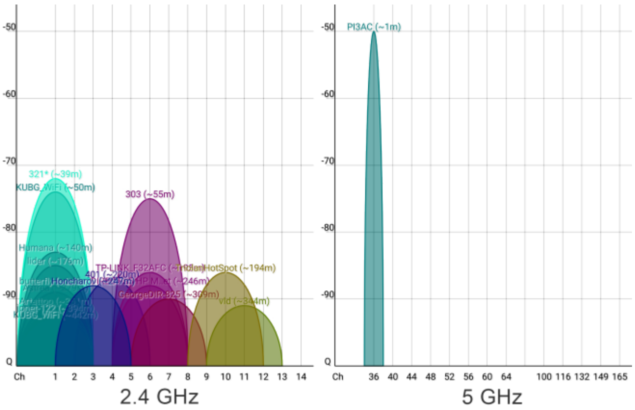

Fig. 6. Radio frequency scan.

VIII. Comparative Analysis of Operating Systems

When measuring and enumeration of the results of the experiment, it is necessary to substantiate its scientific value. To do this, the results obtained were processed and deduced using direct measurement error estimation, download speed, and Pearson coefficient. We got the maximum speed value on the WRT platform with maximum speed of 600 kb/s and the lowest on the RPi external platform 200 kb/s. Other measurements failed due to unstable data transfer. Also, there were bugs in the form of not started loading after the command start, downloading not all the more on several clients and making it impossible to collect the results together. Limit the maximum number of clients on the TP-Link adapter. It was on *OpenWRT* with internal adapter for 7 and 14 clients. The full results were shown in Table II.

TABLE II. Average Research Results

Clients	Platform	Average speed, kb/s	t-distri-bution	Pearson coefficient
7	RPi external	262.6±7.0	2.045	–0.99896
7	RPi internal	309.4±1.8	2.045	–0.99973

7	WRT external	444.9±54.1	2.045	–0.93667
14	RPi internal	131.6±0.3	1.994	–0.99997

IX. Conclusion and Future Work

This work outlines and justifies the current problems of connecting end devices in wireless networks. Available capabilities of embedded systems can support connections from several to several dozen, but it is necessary to carefully select the wireless adapters offered by manufacturers on embedded platforms or develop solutions using additional adapters, antennas, supported encryption technologies, and data transfer protocols. In this work, the built-in network controller has shown itself far better than the external one. According to the experiment, we can note that:

- The statistics indicate possible errors in the operation of the services. Even if a rather universal Python programming language was taken into account, which simulated the load and normal FTP client-server connection.
- The compatibility of network controllers (chips) and the operation of network services OS, although it is quite broad, however, it cannot guarantee its full performance. So when you need to choose the specific equipment, system architecture and services. Needed to complete the testing and prototype setup. Additionally, each service and intermediate node in the system should be provided with sufficient security at all levels of the data transfer through the OSI model.

In next work, we plan to check a data exchange service, database or web server with a certain level of security. And conducting a penetration test, checking the stability of the security measures described above and a general assessment of the vulnerabilities of smart systems.

References

[1] V. Sokolov, A. Carlsson, and I. Kuzminykh, “Scheme for dynamic channel allocation with interference reduction in wireless sensor network,” in *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology*, Oct. 2017, pp. 564–568. DOI: https://doi.org/10.1109/INFOCOMMST.2017.8246463.

[2] T. S. Sobh, “Wi-Fi Networks Security and Accessing Control,” *International Journal of Computer Network and Information Security*, vol. 5, no. 7, pp. 9–20, Jun. 2013. DOI: https://doi.org/10.5815/ijcnis.2013.07.02.

[3] H. Liu, H. Zhang, W. Xu, and Y. Yang, “A New Secure Strategy for Small-Scale IEEE 802.11 Wireless Local Area Network,” *International Journal of Wireless and Microwave Technologies*, vol. 2, no. 4, pp. 21–27, Aug. 2012. DOI: https://doi.org/10.5815/ijwmt.2012.04.04.

[4] M. Durairaj and A. Persia, “ANM to Perceive and Thwart Denial of Service Attack in WLAN,” *International Journal of Computer Network and Information Security*, vol. 7, no. 6, pp. 59–66, May 2015. DOI: https://doi.org/10.5815/ijcnis.2015.06.07.

[5] M. Sruthi and S. Jayanthy, “Development of Cloud Based Incubator Monitoring System using Raspberry Pi,” *International Journal of Education and Management Engineering*, vol. 7, no. 5, pp. 35–44, Sep. 2017. DOI: https://doi.org/10.5815/ijeme.2017.05.04.

[6] Oestoidea. (2017). *IoT Collection*. [Online]. Available: https://github.com/oestoidea/iot.

[7] G. Loyse, Raspberry-Pi Documentation, Release 0.0, Nov 16, 2017. [Online]. Available: https://media.readthedocs.org/pdf/raspberry-piintro/latest/raspberry-pi-intro.pdf.