Neelamani Samal *et al*, International Journal of Computer Science and Mobile Applications,
National Conference on "The Things Services and Applications of Internet of Things",
Gandhi Institute for Education and Technology (GIET) Baniatangi, 23-24 March 2018, pg. 1-6
ISSN: 2321-8363
Impact Factor: 5.515

A PERFORMANCE SURVEY OF OPERATING SYSTEMS IN IoT ENVIRONMENT

Neelamani Samal[1], Pinki Dalai[2]
[1]*Department of Computer Science and Engineering, GIET, Baniatangi, BBSR, neelamani.samal@gmail.com*
[2]*Research Scholar, Department of Computer Science and Engineering, pinki.pihu15@gmail.com*

Abstract: This paper is a comprehensive survey of the various operating system (OS) available for the Internet of Things (IoT) environment. At first this paper introduces the various aspects of the operating systems designed for the IoT environment where the resource constraint plays a huge role and problem for the operation of the general OS designed for the various computing devices and also the devices for which the IoT is implemented. This paper also describes the various OS available for managing the resource constraint IoT environment along with the various platforms each OS supports, the software development kits available for the development of applications in the respective OS along with the various protocols implemented in these OS for the purpose of communication and networking.

Keywords: IDE, IP, SDK, WSN, IoT, Survey on IoT OS.

1. Introduction

With the recent scientific discoveries and technological advancements, computers have become common and the vision of Internet of Things (IoT) is coming to reality; while Operating Systems (OS) may be considered a relatively old field of research. The IoT's unique role in the real life and the crucial reliance on it being critically efficient in various ways for various types of Things and uses in daily life to make life simpler and convenient. The applications and usage of IoT and the need for its specialized OS's along with the slow reaction and the development time for secure communication plays a vital role.

The whole IoT environment is based on the application of microprocessors and wireless sensors. The resource constraint environment of these microprocessors and sensors makes the use of regular OS meaningless due to their high resource and computing power requirement. Thus, in such a situation, the development of OS meeting the resource constraint demand of the IoT environment becomes necessary.

2. OVERVIEW

In section 3 of this paper we introduce the various aspects of an OS designed for the IoT environment. In section 4, the various OS'es available for running in the IoT environment along with a list of the supporting platforms, SDKs and the various networking and communication protocols implemented are surveyed. The paper is concluded in section 5.

3. INTRODUCTION TO OS FOR THE IOT ENVIRONMENT

The Operating System (OS) plays a vital role in managing the content and coordinating among the processes. The OS as a whole maintain a communication and provide a proper security mechanism for data handling. The whole integration of the various IoT devices to the various objects is made possible through the interaction of software at a dynamic level along with the use of wireless sensor network and RFID technologies , Wireless Sensor Network (WSN) using the internet infrastructure ([1], [6]). This software interaction is made possible

ISSN: 2321-8363
Impact Factor: 5.515

through the operating system running in the background within each IoT device without which an IoT device would be nothing more than a non-functioning device. The flexible features of different OS of an IoT device has facilitated some interesting integration of electronic products and technologies to the daily processes of an individual thus making the processes a whole much easier to use and access. Few of the dimensions where the IoT technology has made a major impact are smart light bulb ([28]), implementation of real time passenger information system ([22]), smart tags/NFC tags ([17]), smart houses, smart security etc.

The OS'es developed for the IoT environment small memory generally a few kilobytes of RAM as well as operate with low power consumption. However they are specifically designed for specific applications and are also optimized for a specific set of microprocessor based platforms beyond which such OS becomes irrelevant in its application and may give some unusual value([18]). These OS'es do not compromise in terms of specified features in relevance with the communication, networking, timeliness, security etc. as compared to the regular OS'es like Windows OS, Mac OS etc. but they comes built-in with a number of preinstalled, preintegrated applications, drivers and other network protocols. Moreover these OS'es employ a number of unique security measures for enhancing the IoT infrastructure and also avoid the compromise of the stability and usability of the OS by a common man. Though the security issues of the OS'es for the IoT environment are quite different in comparison to the security issues of a regular operating system, but still it retains the standard security protocols for protecting itself against unwanted attacks. Now a days the IoT environment is made in such a way so as to carry out information exchange between the various electronic devices over the internet in the most efficient way possible using the minimum amount of resources. As of now the whole IoT environment along with its OS becomes vulnerable to malicious attack from the third party intruders. So for the successful implementation of various encryption and data hiding techniques to increase the security measure ([4], [5], [12], [15], [30]), intrusion detection systems (IDS) ([16], [23]) etc. in the IoT infrastructure takes a major importance. [24] and takes care of malicious and hidden attack for making the IoT more secure and robust so that the performance of IoT can be enhanced and be reliably optimised.

4 . OS'ES

i. ARM mbed OS:

The OS is developed by ARM in collaboration with its technological partners, mbed OS is developed for 32-bit ARM Cortex-M microcontrollers ([29]). mbed OS is open source and it can run on a wide range of products ranging from small internet connected devices to smart cites and throughout the smart applications. The whole OS is written using C and C++ language. This open source OS is licensed under Apache License 2.0. This OS require low requirements and good connectivity is going to be a big plus when compared with Microsoft's or Google's offerings.

Some of the key features of the mbed OS are Device and Component Support with Real Time Software Execution and Ease of Use by any client and also provide End to End Security and with a good collection of Drivers and Support Libraries.

The software development kit (SDK) for mbed OS provides the software framework for the developers to develop various microcontroller firmwares to be run on IoT devices. These SDK is comprised of core libraries which consist of the following components given in Table 1:

Networking	Test Scripts	Microcontroller peripheral drivers	RTOS and runtime environment	Built Tools	Debug Scripts

Table 1: Core Libraries in mbed OS

ISSN: 2321-8363
Impact Factor: 5.515

The applications for mbed OS can only be developed online using its native online code editor cum known as mbed online integrated development environments (IDEs). While writing of code can only be done through a web browser, its compilation is done by the ARMCC C/C++ compiler in the cloud platform SaaS. For the connectivity , the mbed OS support the following connectivity technologies given in Table 2:

Bluetooth	Wi-Fi	Zigbee IP	Zigbee LAN	Cellular	Ethernet	6LoWPAN

Table 2: Connectivity technologies in mbed OS

mbed OS integrates end-to-end IP security that may be IPv4 or IPv6 through TLS and DTLS in its communication channels for increased security of the whole OS environment. For developed using C and C++, this open source OS is licensed under LGPL v2.1. The SDKs available for development of applications in RIOT OS are gcc. Moreover the SDK framework supports application programming in C and C++.

ii. Windows 10 for IoT

Like Windows 7 or 8, the newest version of the Microsoft's flagship OS is also available for use in IoTcalled Windows 10 IoT, the OS is available in 3 different editions as given in table 3.

IoT Mobile	IoT Core	Windows 10 IoT Enterprise

Table 3:Widows IoT editions

The first version, called IoT Mobile, is built to run on ARM architecture. Another version known as IoT Core is made to support Raspberry Pi and Intel Atom processors. The third one is called Windows 10 IoT Enterprise and it is packed with most features found in Desktop version of Windows, but it is limited to running a single app. Among all the Windows 10 for IoT could be preferred choice of developers who are already experienced in Visual Studio and Microsoft ecosystem.

iii. RIOT OS

RIOT powers the Internet of Things like Linux powers the Internet. RIOT is a free, open source operating system developed by a grassroots community gathering companies, academia, and hobbyists, distributed all around the world. This version was eveloped by INRIA, HAW Hamburg and FU Berlin initially, RIOT OS is compatible with ARM Cortex-M3, ARM Cortex-M4, ARM7, AVR Atmega and TI MSP430 devices ([8], [24], [31]). Developed using C and C++, this open source OS is licensed under LGPL v2.1. The SDKs available for development of applications in RIOT OS are gcc, valgrind and gdb. Moreover the SDK framework supports application programming in C and C++. RIOT supports most low-power IoT devices and microcontroller architectures RIOT OS supports all the major communication and networking protocols which are tabulated in Table

IPv6	TCP	UDP	CBOR	CoAP	6LoWPAN	CCN-LITE	RPL

Table 4: Networking protocols in RIOT OS

ISSN: 2321-8363
Impact Factor: 5.515

iv. Google Brillo

The Google Brillo was introduced by Google in the year 2015 in Google I/O Conference, Brillo. This is an upcoming Operating System for IoT. Brillo is aimed at specifying developers with experience in making Android Apps which can be utilized to control the devices with the help of a mobile app. The OS is quite lightweight and used Weave protocol for communication purpose.

This Google's Weave communication protocol enables the addition of features like compatibility for Google Assistant. Philips Hue and Samsung Smart Things already work with Weave and the organizations like Belkin, Honeywell, TP-Link, and others are in the process of implementing it.

The different Official Development tools for this are as mentioned in the Table:

SDK	APK	ART	GCM	FCM	NDK

Table 5: Development Tools

v. FreeRTOS:

Developed by Real Time Engineers Ltd., ,the world's leading chip companies over a 14 year period, the FreeRTOS kernel is a market leading real time operating system (or RTOS), and the main theme of the OS is to provide standard solution for microcontrollers and small microprocessors. FreeRTOS provides methods for multiple threads or tasks, mutexes, semaphores and software timers with a tick-less mode for low consumption of resources by the running of the various applications. A tick-less mode is provided for low power applications. Thread priorities are supported. FreeRTOS applications can be completely statically allocated.

This OS is written in C programming and with the addition of a few assembly functions . This open source OS is licensed under modified GPL .

vi. VXWORKS:

The VxWorks is a real time operating system (RTOS) developed by the proprietary software by Wind River. First released in 1987, VxWorks is designed for use in embedded systems requiring real-time, deterministic performance and in many cases, safety and security certification, for industries, such as aerospace and defense, medical devices, industrial equipment, robotics, energy, transportation, network infrastructure, automotive, and consumer electronics, where we can find a large use of this OS. The VxWorks comes with the kernel, middleware, board support packages, which is suitable for development suite

VxWorks supports the following target architectures as specified in table :

ARM	Intel architecture	Power architecture

Table 6: Target Architectures

ISSN: 2321-8363
Impact Factor: 5.515

5. Conclusion

From the above survey it can be observed that all the OS'es for the IoT environment are well equipped with all the major networking and communication protocols, security features as well as optimized for efficient usage of computing power in a resource sharing and managing constraint environment. But still the additional implementation of counter measures to online dictionary attacks in the internet infrastructure used by the IoT environment with the additional emphasis on developing a more robust wireless sensor network will help is making the IoT more reliable and robust with the case of handling communication of message and dealing the case of denial of service attack. These measures will definitely contribute to the protection of user's credentials during online transactions or during the logging inside one's personal account in the cloud and will make the whole IoT environment much secure and more reliable.

References

[1] Hermann Kopetz. "Internet of things." In Real-time systems, pp. 307-323. Springer US, 2011.
[2] http://en.wikipedia.org/wiki/TinyOS
[3] Adam Dunkels, Oliver Schmidt, Niclas Finne, Joakim Eriksson, Fredrik Österlind, Nicolas Tsiftesand Mathilde Durvy. "The Contiki OS: The Operating System for the Internet of Things." (2011).
[4] Sandipan Dey, Ajith Abraham, and Sugata Sanyal. "An LSB Data Hiding Technique Using Natural Number Decomposition." In Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on, vol. 2, pp. 473-476. IEEE, 2007.
[5] Philip P. Dang, Paul M. Chau. "Image encryption for secure internet multimedia applications." Consumer Electronics, IEEE Transactions on 46, no. 3 (2000): 395403.
[6] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic and Marimuthu Palaniswami. "Internet of Things (IoT): A vision, architectural elements, and future directions." Future Generation Computer Systems 29, no. 7 (2013): 1645-1660.
[7] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer and David Culler. "TinyOS: An operating system for sensor networks." In Ambient intelligence, pp. 115-148. Springer Berlin Heidelberg, 2005.
[8] https://github.com/RIOT-OS/RIOT
[9] Vladislav Perelman. "Security in IPv6-enabled wireless sensor networks: An implementation of TLS/DTLS for the Contiki operating system." PhD diss., MSc Thesis, Jacobs University Bremen, 2012.
[10] http://www.nanork.org/projects/nanork/wiki
[11] Vipul Goyal, Virendra Kumar, Mayank Singh, Ajith Abraham, and Sugata Sanyal. "A new protocol to counter online dictionary attacks" Computers &Security, 25, no. 2 (2006): 114-120.
[12] Chi-Kwong Chan, Lee-Ming Cheng. "Hiding data in images by simple LSB substitution." Pattern recognition 37, no. 3 (2004): 469-474.
[13] David J. Malan, Matt Welsh, and Michael D. Smith. "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography." In Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on, pp. 71-80. IEEE, 2004.
[14] Arijit Ukil, Jaydip Sen, and Sripad Koilakonda. "Embedded security for Internet of Things." In Emerging Trends and Applications in Computer Science (NCETACS), 2011 2nd National Conference on, pp. 1-6. IEEE, 2011.
[15] A. V. N. Krishna, S. N. N. Pandit, and A. Vinaya Babu. "A generalized scheme for data encryption technique using a randomized matrix key." Journal of Discrete Mathematical Sciences and Cryptography 10, no. 1 (2007): 73-81
[16] Animesh Kr Trivedi, Rajan Arora, Rishi Kapoor, Sudip Sanyal, and Sugata Sanyal. "A Semi-distributed Reputation Based Intrusion Detection System for Mobile Adhoc Networks." arXiv preprint arXiv: 1006.1956 (2010).
[17] Jeffrey C. Reynar, and Ziyi Wang. "System and method for incorporating smart tags in online content." U.S. Patent 7,003,522, issued February 21, 2006.
[18] Anuj Sehgal. "Using the Contiki Cooja Simulator." (2013).
[19] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors." In Local Computer Networks, 2004. 29th Annual IEEE International Conference on, pp. 455-462. IEEE, 2004.
[20]Benny Pinkas, and Tomas Sander. "Securing passwords against dictionary attacks." In Proceedings of the 9th ACM conference on Computer and communications security, pp. 161-170. ACM, 2002

ISSN: 2321-8363
Impact Factor: 5.515

[21] Lander Casado, and Philippas Tsigas. "Contikisec: A secure network layer for wireless sensor networks under the contiki operating system." In Identity and Privacy in the Internet Age, pp. 133-147. Springer Berlin Heidelberg, 2009.
[22] K. Ganesh, M. Thrivikraman, Joy Kuri, Haresh Dagale, G. Sudhakar, and Sugata Sanyal. "Implementation of a real time passenger information system." arXiv preprint arXiv: 1206.0447 (2012).
[23]Alain Hiltgen, Thorsten Kramp, and Thomas Weigold. "Secure internet banking authentication." Security & Privacy, IEEE 4, no. 2 (2006): 21-29.
[24]Emmanuel Baccelli, Oliver Hahm, M. Gunes, M. Wahlisch, and Thomas C. Schmidt. "RIOT OS: Towards an OS for the Internet of Things." In Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, pp. 79-80. IEEE, 2013.
[25] http://en.wikipedia.org/wiki/FreeRTOS
[26] http://www.freertos.org/
[27] Koustubh Kulkarni, Sudip Sanyal, Hameed AlQaheri, and Sugata Sanyal. "Dynamic Reconfiguration of Wireless Sensor Networks." IJCSA 6, no. 4 (2009): 16-42.
[28] Ihor Lys, and George G. Mueller. "Smart light bulb." U.S. Patent 6,528,954, issued March 4, 2003.
[29] https://mbed.org/technology/os/
[30]Sachin Babar, Antonietta Stango, Neeli Prasad, Jaydip Sen, and Ramjee Prasad. "Proposed embedded security framework for internet of things (IoT)." In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on, pp. 1-5. IEEE, 2011.
[31] http://www.riot-os.org/
[32] Yongzhong He, and Zhen Han. "User authentication with provable security against online dictionary attacks." Journal of Networks 4, no. 3 (2009): 200-207

