International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 10 | Oct 2019 www.irjet.net p-ISSN: 2395-0072

Internet of Things based 6LowPAN Network over Wireless Sensor Network

Mukund Gode[1], Akash Parihar[1], Virat Patel[1], Prof. Khushbu Tikhe[2]

[1]*Student, Department of Electronics Engineering, Shree. L. R. Tiwari College of Engineering, Mumbai, Maharashtra, India*
[2]*Professor, Department Of Electronics Engineering, Shree L. R. Tiwari College of Engineering, Mumbai, Maharashtra, India*

---***---

Abstract - *The Internet of things (IOT) refers to the connection of smart things (motes) to the internet which enables these devices to collect and exchange data and wireless sensor networks (WSN) are at the crux of the Internet of Things applications. In present scenario, there exist many technologies each having its own advantages and disadvantages in IOT. Traditionally, wireless sensor node was not set up using internet because of the power and memory constraints posed by the tiny sensor nodes. The 6LoWPAN technology by IEFT makes possible to use IPv6 communications in sensor networks, which helps solve the problem of interoperability, enabling low power, low cost micro-controllers to be globally connected to the internet. Another constrained application protocol, CoAP allows interactive communication over the internet for these resource constrained devices. With the help of IEEE 802.15.4, 6LowPAN and CoAP protocol, it is possible to implement an open standard WSN model for resource constrained device. The Contiki OS, touted as the open source OS for IoT, provides low power IPv6 communications and supports the 6LoWPAN and CoAP protocols, along with mesh routing using RPL. We have made use of these open standard tools in this work to form an open, inter-operable, scalable, reliable and low power WSN stack. This stack is then simulated using Contiki's default network simulator Cooja, to show how a wireless sensor network is set in IOT, using help of Ipv6 addressing and 6LowPAN, and also show how sensor, sensed information is routed between the sensor nodes and how sensed data can be seen using CoAP and Http application protocols.*

Key Words: IOT, 6LowPAN, Contiki OS, Cooja.

1. INTRODUCTION

Internet of things (IOT) refers to connect things to Internet, things are an entity that has intelligence to sense data around its environment, and send it to the Internet, where it processes the data and make it useful for engineering purpose. The Internet Engineering Task Force, IETF, provides its own description of the Internet of Things, "The Internet of Things is the network of physical objects or "things" embedded with electronics, software, sensors, and connectivity to enable objects to exchange data with the manufacturer, operator and/or other connected devices." [1] All these definitions talk about the networking and data communications along with network services, which are realized by wireless sensor networks. These WSNs must be designed with sophisticated and extremely efficient communication protocols along with sensors with advanced sensing capabilities. This paper attempts to do exactly that, studying the ipv6 and 6lowpan networking protocol, selecting the most optimal operating system i.e. Contiki OS, benchmarking them using simulations i.e. Cooja simulation.

2. BACKGROUND

2.1 IP in WSN

The IP approach was considered unfeasible to operate on microcontrollers and low-power links, mainly due to the introduced header overhead. However, the IETF 6LoWPAN draft standard changes this, defining how IPv6 packets can be efficiently transmitted over IEEE 802.15.4 radio links. 6LoWPAN defines an adaptation layer to carry IPv6 packets over low data rate, low power, small footprint radio networks (LoWPAN) [4]. Making use of IP in WSN by giving a unique IP address to a constrained thing simplifies the connectivity model. This eliminates the need for complex gateways necessary to translate between proprietary protocols and the standard Internet. We can instead use conventional infrastructure such as switches/routers/bridges, which are well understood, well developed and widely available. In IOT OSI stack the implementation of IEEE 802.15.4 protocol is mostly used in physical and data link layer to communicate with sensors and other constrained devices. However, such protocols present interoperability problems when trying to transfer packets to devices outside the IEEE 802.15.4 network, and thus, IP connectivity is desired. IP is open, scalable, lightweight, secure, end to end, stable, versatile and is proving to be ubiquitous in communication technologies today [4]. IP has proved to be a formidable invention and is the backbone of the modern internet communications today. Applying this protocol to WSNs would truly unleash this vision of Internet of Things, where even the smallest of nodes would be connected to the global internet, hence making it an integration protocol.

2.2 IOT Protocol

In current scenario there is lots of research going on for protocol that supports IOT technology and to standardized such protocols. IEEE 802.15.4 is an industry standard specifying the PHY and MAC layers for low-rate wireless personal area networks (LR-WPANs). It has over the years become the de-facto standard for wireless sensor networks and is maintained by the IEEE 802.15 working group [4]. While selecting protocols for wireless communication, its Interoperability with another device protocol in a network should be consider. The ZigBee standard defines all layers from its own network layer right up to the application layer above 802.15.4 PHY and MAC layers. Therefore, ZigBee devices can only communicate with other ZigBee devices. In case of 6LoWPAN, interoperability is ensured as communication is possible with other wireless 802.15.4 devices as well as with devices on any other IP network link (e.g., Ethernet or Wi-Fi) using a border router, which is a simple bridging device. Also, 6LoWPAN doesn't necessarily append additional headers, which brings down packet overhead and allows more payload to be carried. Coming to the transport layer, UDP remains a light weight protocol option with low overheads and complexity with respect to TCP, and hence is the popular option for WSNs but both MQTT and HTTP use TCP, while CoAP employs UDP. Both CoAP and MQTT are open standards, lightweight and suitable to constrained environments than HTTP, allow event-based communication, can have IP underneath and have various implementations. MQTT gives flexibility in communication patterns and acts primarily as a channel for data, whereas CoAP is designed for web interoperability.

3. PROTOCOL STACK

3.1 IPv6

Internet Protocol version 6 addressing, is an upgraded version of IPv4.In IPv6 it uses 128-bit addresses, versus 32-bit addresses used by IPv4. Compared to the total possible number of IPv4 addresses, 4.29 billion, IPv6 provides nearly 600 quadrillion addresses for every square millimeter on earth. That's 6x1023 addresses for every square meter of the earth's surface [2]. This is quite a reason why there is a boom in IOT technology plus it also solve the problem of NAT barrier and supports Multi-Stakeholder and are well suited to devices with constrained resources e.g. 6LowPAN (wireless nets), COAP (transport with web services) and DTLS (secured datagram). Indeed, there is a whole REST environment targeted at constrained devices.

3.2 6LowPAN

6LowPAN is an acronym for IPv6 over Low Power Area Network, is a standard for enabling IP in low power wireless area network. IETF released two RFCs standard of 6LowPAN which are RFC4919 and RFC4944. 6LowPAN provides interoperability with all other potential IP network, while it meets the critical embedded wireless requirement, it provides high reliability and adaptability, long lifetime on limited energy, manageability of many devices and allows link layer mesh routing under IP topology.

IP Protocol Stacks			6LoWPAN Protocol Stacks	
HTTP, SOAP, XMPP		Application	CoAP, DPWS, XMPP	
TCP, UDP	ICMP	Transport	UDP	ICMPv6
IP		Internet	IPv6	
			6LoWPAN Adaptation	
Ethernet		Link	IEEE 802.15.4	

Fig. - 1 illustrates the comparison between IP protocol stack and 6LowPAN protocol stack.

IPv6 payload i.e. minimum of maximum transfer unit MTU 1280 bytes but 802.15.4 carries only 127 bytes so, it's almost x10 bigger than physical layer (hardware layer) so, IPv6 is big protocol that cannot fit into smaller data layer or Physical layer for that we need to have adaptation layer. Hence 6LowPAN is used fragment mechanism under IP layer and compression mechanism (HC1, HC2) to compress ipv6 header. HC1 used to compress network layer header and HC2 used to compress transport layer header.

4. SYSTEM ARCHITECTURE

Basically, idea is to set a 6LowPAN based mesh network between the sensor nodes, sensor node can be any microcontroller or microprocessor that has support of IPv6 and 6LowPAN protocols e.g. Tmote Sky mote, Wismote mote, Z1 mote etc. Each of these motes in a network will be programmed as websense mote, and will have a unique IPv6 address and will be configured with RPL routing algorithm. One of the mote will be configured as Border Router, the border router or edge router is typically a device sitting at the edge of our network, which allows us to outside network using its build in network interface such as Wi-Fi, ethernet, Serial etc. Each mote will sense, parameter within it environment using a sensor embedded on it, and will pass the sense data to the edge router, using single hop, if the mote is within the range of edge router or by multi hop if it is outside the range of border router, will routes it path as per RPL routing algorithm. Edge router also provide tunneling of IP address from IPv6 to IPv4.

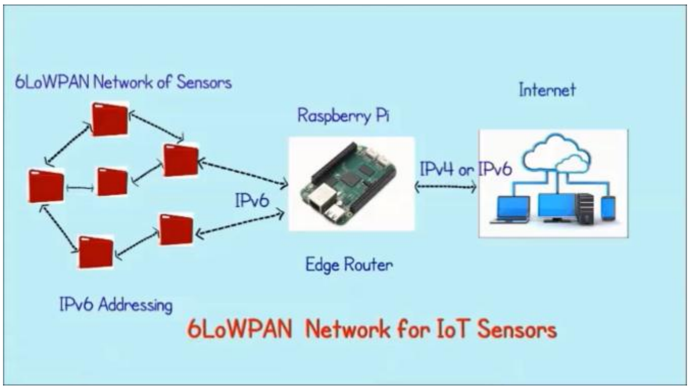

Fig. 2 – System Architecture

5. TOOLS AND TECHNOLOGIES

In this paper we try to make use of tools and technologies that gives interoperability, reliability, simplicity, availability, low power consumption, low memory, low cost etc. This work relies on open standard.

5.1 Contiki OS

An open source IOT OS that connects tiny low power, low cost microcontrollers to the internet, it is also powerful toolbox for building complex wireless systems and as it is open source there is no need to write code, full source code is available in one instant Contiki download. The Contiki OS provides Low Power Internet Communication, it supports fully standard ipv4 and ipv6, along with recent low-power standards: 6LOWPAN, RPL, CoAP etc.

Contiki supports IP networking through the uIP TCP/IP stack. Contiki also has support for IP routing using the RPL protocol that has been standardized by the IETF (RFC 6550 from Roll WG). 6LoWPAN border router, provides low-power mesh network with connection to external networks (Internet, ..).

5.2 Cooja simulator

It helps the programmers to see their applications run in large scale networks or in high detail on fully emulated hardware devices. Contiki devices build complex wireless network. So, developing and debugging programs in such a network is difficult so cooja helps in providing simulation environment where developers can easily debug their software. Contiki/tools/cooja/

6. SIMULATION

Simulations do not suffer from NAT/ firewall issues, interconnectivity issues of IPv6/ IPv4, environmental interference and noise etc. They offer an ideal model of working which tries to simulate protocols as closely as possible to their recommendations and formats, to understand their behavior and structure.

6.1 Simulation scenario

Simulation is design to show how 6LowPAN network is set over Wireless sensor network using Contiki OS on Cooja simulator.

In Simulation we had made use of Tmote Sky mote in which one sky mote is programmed as rpl border router and every other node as websense node to sense the environmental parameter. The code of the border-router is in path Scontiki/examples/ipv6/rpl-border-router/border-router.c. The program for websense node is in ipv6/sky-websense/sky-websense.c.

From the Network window, right click on the border-router and select "Mote tools for sky/Serial socket (SERVER)" from the menu. A new window should appear saying that the border-router node is listening on local port 60001 [3]. When your simulation is ready to start, do the following in another terminal window:

cd to your project folder where the rpl-border-router is located

$ make connect-router-Cooja TARGET=sky

This will start a program named tunslip6 that sets up an interface on the Linux IP stack and connects this interface via a socket to the border router node in Cooja.

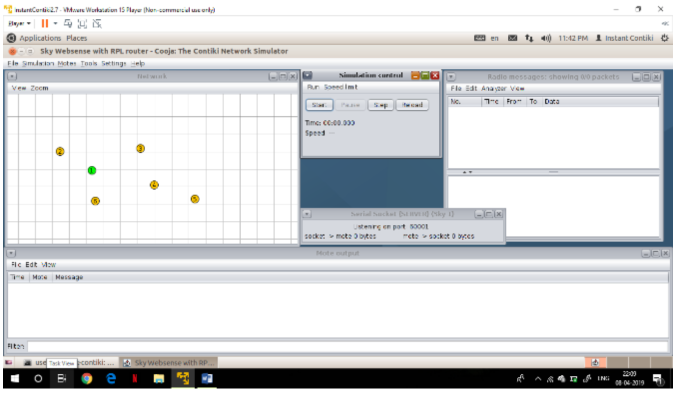

Fig. 3- Sensor motes placed on Simulation Screen

In this work we have added 5 websense node and one border-router node for the simulation.

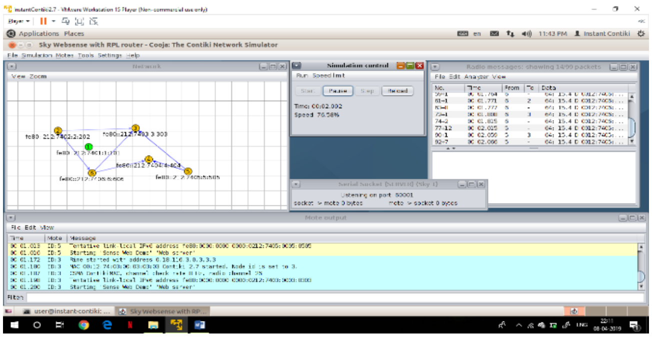

Fig. 4 – Sensor node routes to transfer data

In this it is seen that, each node is assigned with a unique IPv6 address. In a network each node is identified using this IPv6 address. Each websense node make a connection we a border router to make a connection with external network.

Once everything is set, you can start your simulation. We can see about, how many websense node is connected to border router, how they are routing their packets to border router in a network on your desktop by using IP address of your border router.

http://[aaaa::212:7401:1:101]/

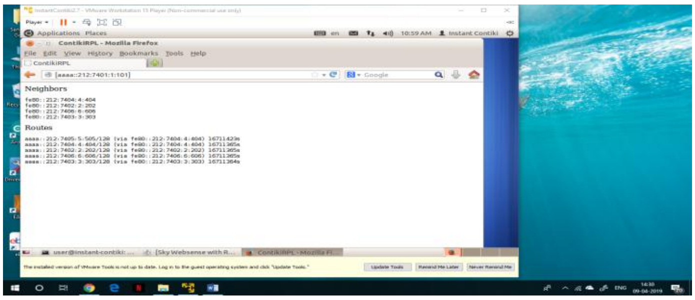

Fig. 5 – Border Router Information shown on Http Application

We in this simulation are using Http application protocol but for IOT application we usually prefer Coap application protocol because of its advantages for IOT setup.

On the sky-websense node, the sensors enabled for this test are temperature and light. You can consult the webserver by searching for http://[‘‘webserver-ipv6-addr’ ’] in your web browser to see the available readings.

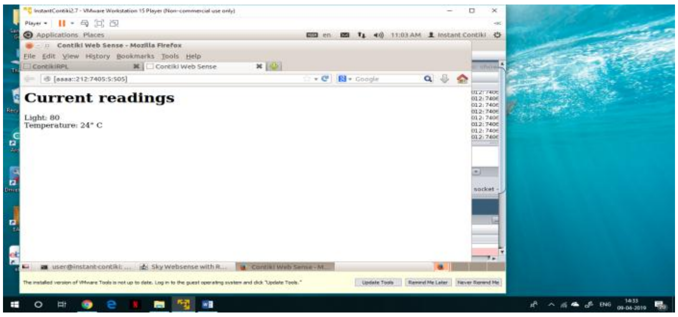

Fig. 6 – Sensor data from Websense node

7. Conclusion

In this paper we have shown, how IPv6 based networking for Internet of things over wireless sensor network using Contiki OS and Cooja simulation. We made use of 6LowPAN protocol as adaption layer for IPv6 and used RPL protocol, which is routing protocol for low power and lousy networks for routing purpose.

REFERENCES

[1] Internet Engineering Task Force (IETF). https://www.ietf.org/topics/iot/ .

[2] IPv6 The furture of Internet https://www.ipv6.com/general/ipv6-the-future-of-the-internet/

[3] Ing. Pietro Gonizzi, Dr. Simon Duquennoy. Hands on Contiki OS and Cooja Simulator: Exercises (Part II). 17-09-2013

[4] Sumeet Thombre, Raihan Ul Islam, Karl Andersson, and Mohammad Shahadat Hossain. “IP based Wireless Sensor Networks: Performance Analysis using Simulations and Experiments.” Pervasive and Mobile Computing Laboratory Lulea University of Technology SE-931 87 Skelleftea, Sweden.