Tunneling 6LoWPAN protocol stack in IPv6 network

Cite as: AIP Conference Proceedings **1755**, 070006 (2016); https://doi.org/10.1063/1.4958501
Published Online: 21 July 2016

Alfadoni Prisantama, Widyawan, and I. Wayan Mustika

Tunneling 6LoWPAN Protocol Stack in IPv6 Network

Alfadoni Prisantama[1, a)], Widyawan[1, b)], I Wayan Mustika[1, c)]

[1] *Dept. of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Yogyakarta, Indonesia*

[a)] Corresponding author: alfadoni.prisantama@mail.ugm.ac.id
[b)] widyawan@ugm.ac.id
[c)] wmustika@ugm.ac.id

Abstract. Nowadays monitoring systems are designed with embedded technology such as Wireless Sensor Network. A communication stack based on IPv6 for low power devices named 6LoWPAN gives a great opportunity for sensor networks to communicate with the Internet using IP. However, with a few differences between the 6LoWPAN protocol stack and IP protocol stack, it is difficult for 6LoWPAN sensor networks to be operated from the Internet directly. In order to be directly connected to IPv6 networks, tunneling is required. Therefore, tunneling process test on 6LoWPAN network to connect to IPv6 networks needs to be done to show how each 6LoWPAN nodes will be accessible through the edge node and IPv6 network. In this research, we have conducted experiments to design the 6LoWPAN networks and test the tunneling process on 6LoWPAN network to connect with IPv6 networks. We have created a 6LoWPAN network with one edge node and two end nodes. The result shows that each node will be accessible using its IPv6 address through the edge after tunneling process is successfully executed. This research can become an insight for further studies on 6LoWPAN protocol stack as a part of the WSN-based monitoring system in movement.

INTRODUCTION

Nowadays monitoring systems are designed with embedded technology such as Wireless Sensor Network (WSN). ZigBee is one of the wireless access medium standard for WSN, but the interoperability issue remains a challenge. Meanwhile, the number of IP-enabled devices is increasing very rapidly so that IPv4 addresses are not enough anymore. It led to the need for migration to IPv6 because it provides much more address space. However, because of its size big enough, some adjustments to the sensor nodes need to be made. It led to a creation of open source protocol stack called 6LoWPAN. 6LoWPAN stands for IPv6 over Low-power Wireless Personal Area Network, a communication stack based on IPv6 protocol for low power devices [1].

6LoWPAN gives a great opportunity for sensor networks to communicate with the Internet using IP. However, with a few differences between the 6LoWPAN protocol stack and IP protocol stack, it is difficult for 6LoWPAN sensor networks to be operated from the Internet directly. In order to be directly connected to IPv6 networks, tunneling is required. This tunneling is different from the tunneling on IPv6 transition because it is used for allowing a 6LoWPAN protocol to run over IPv6 network, not providing a way to utilize an existing IPv4 routing infrastructure to carry IPv6 traffic [2].

There are several previous research about tunneling on 6LoWPAN that had been done before, such as using tunnel mechanisms like ISATAP, Teredo, and 6to4, as a method to use 6LoWPAN in IPv4 network [3], tunneling to transmit 6LoWPAN IPv6 packages over IPv4 through a 6to4 tunnel implementation [4], transform the 6LoWPAN packets into IPv6 packets then into IPv4 packets to connect 6LoWPAN network with the Internet [5], an IPv6/IPv4 tunneling application to ensure the tunneling between the nodes, the border routers and the central server [6], using 6to4 tunneling technique to develop Ship Area Sensor Networks (SASN) model which is integration of wired network and 6LoWPAN [7], and automatic tunneling mechanism based on AYIYA by using a public tunnel broker service to connect Android IPv6 smartphone with 6LoWPAN wireless sensors [8].

Advances of Science and Technology for Society
AIP Conf. Proc. 1755, 070006-1–070006-7; doi: 10.1063/1.4958501
Published by AIP Publishing. 978-0-7354-1413-6/$30.00

Unfortunately, most previous research about tunneling on 6LoWPAN was conducted to enable transmit IPv6 packages over IPv4, not tunneling that is used for allowing 6LoWPAN protocol to run over IPv6 network. Most previous research assumed that 6LoWPAN network would run over IPv6 network automatically while the truth there is another process, which is tunneling. Therefore, this tunneling process test needs to be done to show how each 6LoWPAN nodes will be accessible through the edge node and IPv6 network.

The purpose of this paper is to provide an overview of 6LoWPAN and its tunneling process test to connect with IPv6 networks. By understanding 6LoWPAN protocol stack and its tunneling mechanism, this research can become an insight for further studies on 6LoWPAN protocol stack as a part of the WSN-based monitoring system in movement. In this paper, we present an overview of 6LoWPAN protocol stack and its tunneling process.

6LoWPAN PROTOCOL STACK AND ARCHITECTURE

6LoWPAN is a protocol specification to enable IPv6 standards to be used in low-power wireless networks, specifically with IEEE 802.15.4. The IETF 6LoWPAN working group maintains it. The rationale for introducing 6LoWPAN is that the existing IPv6 is too bulky for WSN. In 6LoWPAN, the IPv6 header is compressed to only a few bytes by introducing an adaptation layer that resides between network and MAC/PHY layer while retaining the main IPv6 functionality. The transmission of 1280 bytes IPv6 Maximum Transmission Unit (MTU) over IEEE 802.15.4 is also made possible using fragmentation and reassembly provided by this adaptation layer. The detail specification of this protocol is described in IETF standard RFC4944 [9].

The 6LoWPAN architecture is made up of low-power wireless area networks (LoWPANs), which are IPv6 sub-network. It means a LoWPAN is the collection of 6LoWPAN nodes, which share a common IPv6 address prefix (the first 64-bits of an IPv6 address). LoWPAN nodes may play the role of host or router, along with one or more edge routers, as seen in Fig. 1.

There are three types of LoWPANs which are Simple LoWPANs, Extended LoWPANs, and Ad hoc LoWPANs [10]. A Simple LoWPAN is connected through one LoWPAN Edge Router to another IP network. An Extended LoWPAN consists of multiple edge routers along with a backbone link to interconnect them. An Ad hoc LoWPAN is not connected to the Internet and operates without an infrastructure.

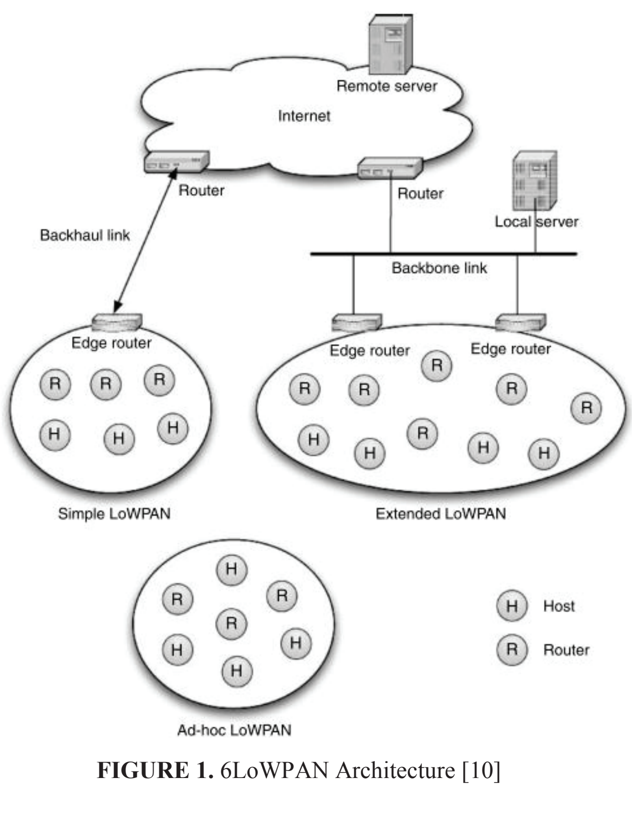

FIGURE 1. 6LoWPAN Architecture [10]

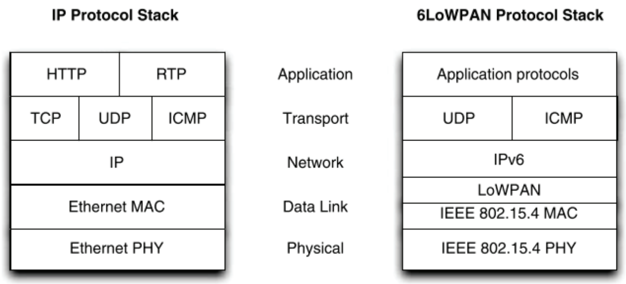

FIGURE 2. 6LoWPAN Protocol Stack [10]

Figure 2 shows the 6LoWPAN protocol stack compared to the IP protocol stack. It is almost identical to a normal IPv6 implementation with the following two differences:

- 6LoWPAN only supports IPv6, for which a small adaptation layer (LoWPAN) has been defined to optimize IPv6 over link layers.
- Although 6LoWPAN is not bound to the IEEE 802.15.4 standard, it is designed to utilize it.

EXPERIMENTAL SETUP

6LoWPAN system architecture consists of 6LoWPAN sensor nodes/end nodes, 6LoWPAN edge node, PC, IPv6 server, and IPv6 clients, as shown in Fig. 3. However, our tunneling experimental setup only consists of 6LoWPAN end nodes, 6LoWPAN edge node, and PC, as shown in the dashed box in Fig. 3.

Our implementation of 6LoWPAN protocol stack is based on Waspmote Mote Runner provided by IBM and Libelium. IBM Mote Runner SDK is used on top of Libelium Waspmote sensor platform to create 6LoWPAN/IPv6 connectivity. IBM Mote Runner SDK is a run-time platform and development environment for wireless sensor networks (WSN) currently under development at the IBM Zurich Research Laboratory, Switzerland. IBM Mote Runner SDK to be installed is the IBM Mote Runner beta 13.1 for Libelium IBM IoT Starter Kit. In order to be able to use IBM Mote Runner SDK, it is required Linux Ubuntu Desktop 12.04 (64-bit) Operating System, Mono JIT compiler version 2.10.8.1, and Java JRE/JDK version 1.6.0_35.

Hardware for edge node and end nodes are Waspmote Mote Runner Gateway and Waspmote Mote Runner Sensor Node. Fig. 4(a) is Waspmote Mote Runner Sensor Node. It has a functionality to send sensed data to edge node and response for each ping message. Fig. 4(b) is Waspmote Mote Runner Gateway. Waspmote Mote Runner Gateway will be combined with PC for 6LoWPAN-IPv6 tunneling. It has a functionality to enable the forwarding of sensing data between 6LoWPAN network and IPv6 external network. We can see that the difference between Waspmote Mote Runner Gateway and Waspmote Mote Runner Sensor Node is on the Ethernet module on the Waspmote Mote Runner Gateway.

We created 6LoWPAN network with one edge node and two end nodes. One node is configured as the edge node, and two other nodes will be the end nodes. 6LoWPAN end nodes should be accessible by IPv6 client from the Internet through the edge node. This is why tunneling is required. Configuration for each node can be seen in Fig. 5. The mrv6-edge library is installed into edge node and the mrv6-lib library is installed into end nodes.

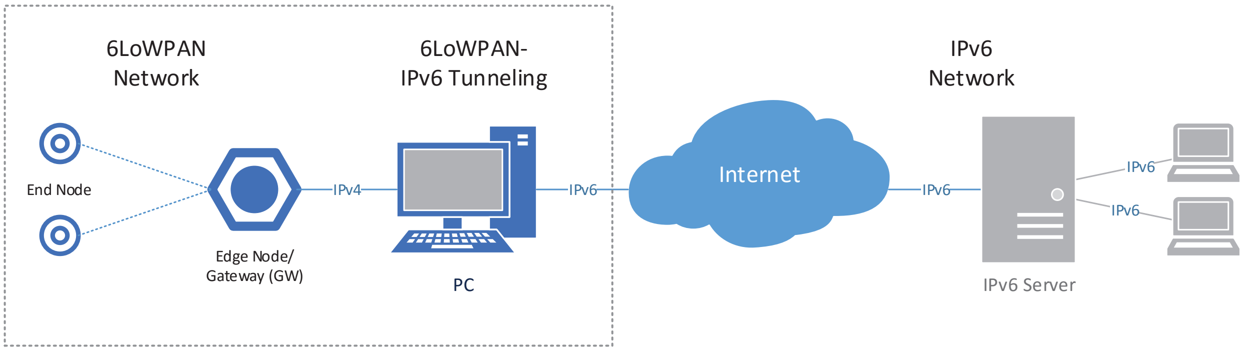

FIGURE 3. 6LoWPAN System Architecture

(a) (b)

FIGURE 4. Waspmote Mote Runner Hardware [11]

6LoWPAN TUNNELING PROCESS

In this section, we will discuss the 6LoWPAN-IPv6 tunneling process on the 6LoWPAN networks to connect with IPv6 network. In this experiment, we set up the 6LoWPAN networks, test the tunneling process, and do the connectivity test. Tunneling is done in the edge node. The tunnel sets up a virtual IPv6 interface to be configured. It will be used to route information between the edge node and IPv6 networks. This tunnel will work as follows:

1. First starts the tunnel to be able to create a virtual interface.
2. Then start the tunnel for edge node.
3. After having started the tunnel, set up the route for the interface.
4. IPv6 address default configuration for nodes is fc00:0db8:0005:0000:<EUI-64>, where EUI-64 is the MAC address of the nodes.
5. After that, the tunnel will generate IPv6 address automatically if there is a new node connected to the edge node.
6. Nodes can be reached using fc00:0db8:0005:0000:**<EUI-64>**, which is:
    - fc00:0db8:0005:0000:**0200:0000:a0f7:3d17** for edge node
    - fc00:0db8:0005:0000:**0200:0000:b93e:b913** for end node I
    - fc00:0db8:0005:0000:**0200:0000:0af4:3d33** for end node II

Process 1 through 4 can be seen in Fig. 6 while process 5 and 6 can be seen in Fig. 7. With the existence of tunnel, IPv6 mapping becomes seamless for the IPv6 clients and nodes. Both IPv6 clients and nodes can be reached and identified using IPv6 addresses. Each node will be accessible using its IPv6 address through the edge node that is connected to the tunnel.

```
moma-list
Mote-Id                  Outdated  State  Assembly         Id  Version
----------------------------------------------------------------------------
02-00-00-00-A0-F7-3D-17            OK     saguaro-system   0   11.4.34896
                                   OK     waspmote-system  1   14.0.34889
                                   OK     mrv6-edge        3   1.0.06930
----------------------------------------------------------------------------
02-00-00-00-B9-3E-B9-13            OK     saguaro-system   0   11.4.34896
                                   OK     waspmote-system  1   14.0.34889
                                   OK     mrv6-lib         2   1.0.06923
                                   OK     sense            5   1.0.13677
----------------------------------------------------------------------------
02-00-00-00-0A-F4-3D-33            OK     saguaro-system   0   11.4.34896
                                   OK     waspmote-system  1   14.0.34889
                                   OK     mrv6-lib         2   1.0.06923
```

FIGURE 5. 6LoWPAN Nodes Configuration

```
Tunnel: opened tunnel device 4, waiting for interface configuration...
New node: 02000000a0f73d17 0
Nodes Full-Addr                                S-Addr       Parent
0000:0000:0000:0000:0200:0000:a0f7:3d17           0         65535
Updated node: 02000000a0f73d17 0
Nodes Full-Addr                                S-Addr       Parent
0000:0000:0000:0000:0200:0000:a0f7:3d17           0         65535
Tunnel: interface ipaddr: fc000db80005000000000000000000ff, xaddr: 0000000000000
0ff
Tunnel: network prefix: fc00:0db8:0005:0000
Tunnel IP interface configured.

Nodes Full-Addr                                S-Addr       Parent
fc00:0db8:0005:0000:0200:0000:a0f7:3d17           0         65535
```

FIGURE 6. Tunnel Process for Edge Node

```
New node: 02000000b93eb913 1
Nodes Full-Addr                                S-Addr       Parent
fc00:0db8:0005:0000:0200:0000:a0f7:3d17           0         65535
fc00:0db8:0005:0000:0200:0000:b93e:b913           1             0
New node: 020000000af43d33 2
Nodes Full-Addr                                S-Addr       Parent
fc00:0db8:0005:0000:0200:0000:a0f7:3d17           0         65535
fc00:0db8:0005:0000:0200:0000:b93e:b913           1             0
fc00:0db8:0005:0000:0200:0000:0af4:3d33           2             0
Updated node: 02000000a0f73d17 0
Nodes Full-Addr                                S-Addr       Parent
fc00:0db8:0005:0000:0200:0000:a0f7:3d17           0         65535
fc00:0db8:0005:0000:0200:0000:b93e:b913           1             0
fc00:0db8:0005:0000:0200:0000:0af4:3d33           2             0
Updated node: 02000000b93eb913 1
Nodes Full-Addr                                S-Addr       Parent
fc00:0db8:0005:0000:0200:0000:a0f7:3d17           0         65535
fc00:0db8:0005:0000:0200:0000:b93e:b913           1             0
fc00:0db8:0005:0000:0200:0000:0af4:3d33           2             0
Updated node: 020000000af43d33 2
Nodes Full-Addr                                S-Addr       Parent
fc00:0db8:0005:0000:0200:0000:a0f7:3d17           0         65535
fc00:0db8:0005:0000:0200:0000:b93e:b913           1             0
fc00:0db8:0005:0000:0200:0000:0af4:3d33           2             0
```

FIGURE 7. Tunnel Process for End Nodes

NODE CONNECTIVITY TEST

Connectivity reflects the network available or not. Ping for IPv6 command then can be used to test the connectivity, such as:

- ping6 fc00:0db8:0005:0000:0200:0000:a0f7:3d17 to test if the edge node is reachable
- ping6 fc00:0db8:0005:0000:0200:0000:b93e:b913 to test if the end node I is reachable
- ping6 fc00:0db8:0005:0000:0200:0000:0af4:3d33 to test if the end node II is reachable

As shown in Fig. 8 and Fig. 9 we send an ICMPv6 message to ensure that the nodes are completely reachable. Figure 8 shows that edge node is reachable and Fig. 9 shows that end nodes are reachable. It appears that edge node and end nodes are already reachable. The result shows that our system can realize 6LoWPAN network and IPv6 address is already generated. It means the tunneling process is successfully executed. Now information can be sent directly from the nodes to the IPv6 clients and vice versa.

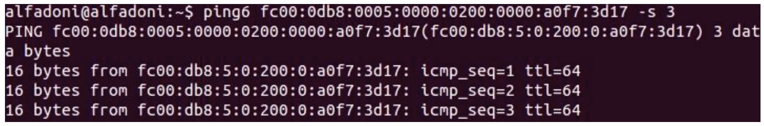

FIGURE 8. Connectivity Test for Edge Node

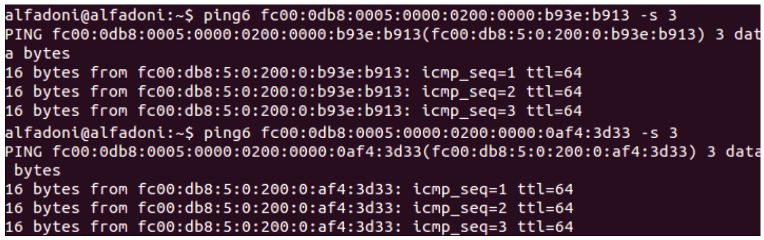

FIGURE 9. Connectivity Test for End Nodes

CONCLUSION

In this paper, we have conducted experiments to design the 6LoWPAN networks and test the tunneling process on 6LoWPAN network to connect with IPv6 networks. We have created a 6LoWPAN network with one edge node and two end nodes. The tunneling experiments had been done for implementation design and connectivity test. From the test, we can find that the tunnel sets up a virtual IPv6 interface to route information between the edge node and IPv6 networks. Therefore, each node will only be accessible through the edge node that is connected to the tunnel. The experiment results can become an insight for further studies on 6LoWPAN protocol stack as a part of the WSN-based monitoring system in movement. Because of the importance of the tunneling for 6LoWPAN to connect with IPv6 network, we recommend in the future this work focusing on improving the tunneling process and analyzing the performance of 6LoWPAN protocol stack.

ACKNOWLEDGMENTS

This work was supported by Directorate of Systems and Information Resources, Universitas Gadjah Mada through project 6LoWPAN-based Campus Monitoring.

REFERENCES

1. N. Kushalnagar, G. Montenegro, and C. Schumacher, *RFC 4919 : "6LoWPAN : Overview, Assumptions, Problem Statement, and Goals"* (IETF, California, USA, 2007), pp. 1–12.
2. R. Gilligan and E. Nordmark, *RFC 2893 : "Transition Mechanisms for IPv6 Hosts and Routers"* (IETF, California, USA, 1996), pp. 1–29.
3. C. Yum, Y. Beun, S. Kang, and Y. Lee, "Methods to use 6LoWPAN in IPv4 network," in *International Conference on Advanced Communication Technology* (IEEE, Phoenix Park, Korea, 2007), pp. 969–972.
4. G. Yu, H. Min, H. Wei, and H. Huang, "Design and Implementation of Interconnecting IPv6 Wireless Sensor Networks with the Internet," in *International Conference on Robotics and Biomimetics* (IEEE, Guangzhou, China, 2012), pp. 1325–1330.
5. Y. Xu, R. Tan, S. Wu, and Q. Tang, J. Networks **8**, 1480 (2013).
6. M. Carlier, A. Braeken, B. Lemmens, R. Smeets, and L. Segers, "6LoWPAN – Towards Zero-Configuration For wireless Building Automation: System Architecture," in *International Electronic Conference on Sensors and Applications* (MDPI, 2014), pp. 1–6.
7. S.J. Nam, J. Kang, and D. Moon, "Adaptation of 6LoWPAN to Ship Area Sensor Networks with wired Ethernet backbone and performance analysis," in *14th International Conference on Control, Automation and Systems* (IEEE, Yongin, South Korea, 2014), pp. 967–970.
8. P. Schleiss, N. Tørring, S.A. Mikkelsen, and R.H. Jacobsen, "Interconnecting IPv6 Wireless Sensors with an Android Smartphone in the Future Internet," in *Baltic Congress on Future Internet Communications* (IEEE, Vilnius, Lithuania, 2012), pp. 14–18.

9. N. Kushalnagar, G. Montenegro, D.E. Culler, and J.W. Hui, *RFC 4944: Transmission of IPv6 Packets over IEEE 802.15.4 Networks* (IETF, California, USA, 2007), pp. 1–30.
10. Z. Shelby and C. Bormann, "The 6LoWPAN Architecture," in *6LoWPAN: The Wireless Embedded Internet*, edited by D. Hutchison, S. Fdida, and J. Sventek, 1st ed. (Wiley, New York, USA, 2009), pp. 13–17.
11. Libelium, "Waspmote's Hardware," in *Waspmote Mote Runner Technical Guide*, edited by Libelium Dev Team and IBM Dev Team (Libelium, Zaragoza, Spain, 2015), pp. 21–29.