Research Article

A Study of BLE Mesh Network Scheduling Algorithm Using A Smartplug

Goo Kim[1] and Hyungyoon Seo*

[1] Research Professor, The Korea Ship and Offshore Research Institute, Pusan National University, 2 Busandaehak-ro, 63 beon-gil, Geumjeong-gu, 46241, Rep. of Korea

* Assistant Professor, Dept. Of Computer Software Engineering, Changshin University, 262 Paryong-ro, MasanHoiwon-gu Changwon-si, Gyeongsangnam-do, 51352, Rep. of Korea

*Corresponding author. Tel.: +82-055-250-1313; Email address: hyseo@cs.ac.kr

Article History:Received:11 november 2020; Accepted: 27 December 2020; Published online: 5 April 2021

Abstract: This paper proposes a BLE Mesh network scheduling algorithm using SmartPlug. And proposes random-backoff, a very simple method for collision avoidance. The low-energy scheduling algorithm of SmartPlug and BLE node is proposed. SmartPlug periodically broadcasts its own information to inform the surrounding BLE nodes and SmartPlug. A node for data communication with SmartPlug is decided for efficiency. The BLE node periodically transmits data and switches to sleep mode after data transmission. In this paper, propose a very simple method, random-backoff, to reduce collisions when transmitting data in a BLE node. In the simulation of results shows when the data size is 1 ~ 31 bytes and 32 ~ 255 bytes, and random-backoff is effective when 1~31 bytes length. In the case of 32 to 255 bytes, random-backoff is not effective, so additional research is needed. And the simulation conducts under the same condition, but it shows that the data length and the number of transmission attempts has an effect. The simulation results show the transmission success ratio is similar regardless of the data length when random-backoff is not applied. It also shows the number of transmission attempts has an effect. Also, this simulation shows the results that as the number of nodes increases, the wireless environment becomes congested, and the transmission success ratio decreases. As a result of the simulation, random-backoff for collision avoidance is effective in transmitting data of 1 to 31 bytes better than 32 to 255 bytes in the data length.

Keywords: Smart Factory, BLE, BLE Mesh, Scheduling Algorithm, Collision Avoidance

1. Introduction

This paper proposes a BLE Mesh network [1] scheduling algorithm using SmartPlug. And proposes random-backoff, a very simple method for collision avoidance. In this paper, SmartPlug is a device that can sense the data such as the harmful gas and transmits sensing data to a smart factory management system in real-time. The sensing data such as the harmful gas is transmitted through IoT [2] communication device like a BLE [3, 4] Mesh node. So, we insert a BLE Mesh node into a SmartPlug. BLE has improved communication speed, low power efficiency, and a one-hop communication distance compared with Bluetooth 4.2 [5]. So, we expect BLE can be used for a variety of environments like Smart Home and Smart Factory. The data transmission method of the BLE Mesh node is Managed Flooding, and switch to sleep mode after transmitting a data. Therefore, there are Friend Nodes (SmartPlugs in this paper) in BLE Mesh networks for data relay, and do not switch to sleep mode. A BLE Mesh node awakes according to the set by the administrator and transmits data to the friend node. However, a collision occurs when another BLE Mesh node awakes and transmits a data at the same time. Therefore, it needs a scheduling algorithm to control the BLE Mesh nodes. Also, in this paper propose random-backoff, a very simple method for collision avoidance. BLE node does frequency hopping. This paper assumes that the BLE node sends the data through 3 broadcasting channels. This paper shows the results of simulations that increase the number of BLE nodes is from 2 to 4 and divide the data length is 1 ~ 31 bytes and 32 ~ 255bytes. The simulation results better when applying random-backoff to the data length 1 ~ 31 bytes. This paper shows the result of simulation that random-backoff is no effect when the data length longer than 31 bytes.

2. Related review

Energy-efficient scheduling on wireless embedded devices is highly important due to constraint energy and data processing time. If the administrator wants low-energy consumption on the BLE node, it can set a long interval for the data transmission interval. However, it is difficult to transmit data in constraint time. In contrast, it is difficult to realize the low-energy consumption of the BLE node if the administrator sets short the data transmission interval. It controls the parameters for the low-energy consumption of the BLE node. The master node receives the data length and the transmission interval from BLE nodes. The master node schedules the data transmission timing for transmission packet collision avoidance of the BLE nodes [6].

If there are many BLE devices in the local area, there is possible interference with each other during the data transmission. BLE does frequency hopping, it is meant to change the frequency when data transmission. However, if there are many IoT devices like BLE, Wi-Fi, and so on in the local area, they can be a possible collision when transmits data, and data can be the loss. In this paper [7], it analyzes the packet collision probability according to the event interval and the throughput. If the condition of the wireless environments may become congested, the data may be lost in BLE mesh networks when the data are transmitted. BLE mesh environments in Software Defined Networking (SDN) are then constructed to transmit data [8], may detect the

***Corresponding author:** Hyungyoon Seo

Assistant Professor, Dept. Of Computer Software Engineering, Changshin University, 262 Paryong-ro, MasanHoiwon-gu Changwon-si, Gyeongsangnam-do, 51352, Rep. of Korea

Email address: hyseo@cs.ac.kr

congestion of wireless network environment, and adjust the transmission parameters to prevent the overflow of buffer and maintain the performance of the transmission. Also, it prevents the delay of the transmission. In related papers [8, 9], it controls the BLE network congestion by the controls the throughput. The wireless network configuration may vary depending on the degree of radio congestion in the real industrial field. It is already been studied about the coexistence issue of different types of wireless network systems. In related paper [10, 11, 12] studies about 2.4GHz band interference using in Wi-Fi, Bluetooth, and WiMAX. In related paper [10] imparts probabilistic treatment presented in the slotted p-Persistent CSAM/CA on a co-existing Bluetooth piconet. It shows packet error probability in Bluetooth and Bluetooth throughput according to transmission probability and slot duration. The related paper [11] proposes a non-collaborative coexistence mechanism for wireless-fidelity and Bluetooth systems based on dynamic packet fragmentation.

In this paper, SmartPlug receives the information from BLE nodes in around a SmartPlug and schedules the data of the BLE nodes transmission interval. SmartPlug can schedule the transmission timing of other BLE nodes for collision avoidance.

3. Proposed Work

3.1 Plow-energy scheduling algorithm

This section proposes a low-energy scheduling algorithm using SmartPlug in the BLE Mesh network. Figure 1 shows the low-energy scheduling algorithm of the SmartPlug and the BLE nodes.

The SmartPlug operations are as follows. A SmartPlug periodically broadcasts its own information to inform the surrounding BLE nodes and SmartPlugs. Simultaneously, a SmartPlug collects and stores the received information from around the BLE nodes and SmartPlugs. A SmartPlug does not need to communicate with all the collected BLE nodes because it is inefficient. For example, if a BLE node can communicate to SmartPlug-A better than SmartPlug-B, it chooses to communicate with SmartPlug-A. Therefore, a SmartPlug determines the BLE nodes to communicate with by transmitting and receiving BLE node information between SmartPlugs. SmartPlug sets the data transmission interval for each node and transmits it to the nodes. Finally, a SmartPlug floods the data received from the nodes. At the same time, performs the channel hopping for avoiding collision of data transmission, and data is flooding after random-backoff for avoiding collision of data transmission even after channel hopping.

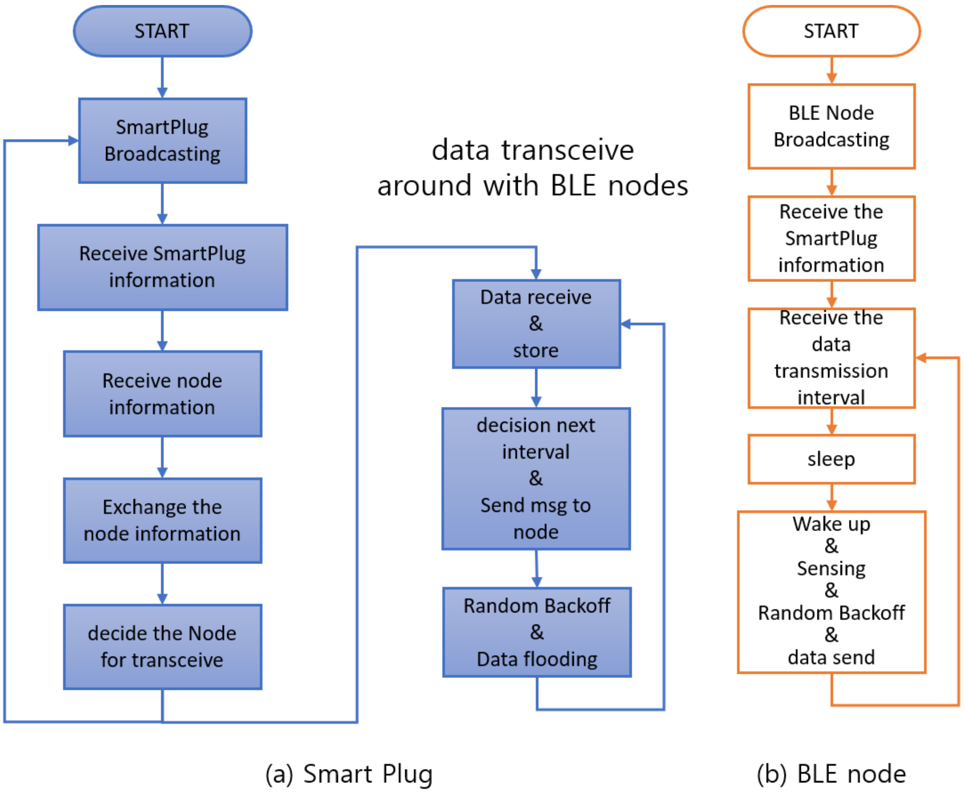
START
SmartPlug Broadcasting
Receive SmartPlug information
Receive node information
Exchange the node information
decide the Node for transceive
data transceive around with BLE nodes
Data receive & store
decision next interval & Send msg to node
Random Backoff & Data flooding
START
BLE Node Broadcasting
Receive the SmartPlug information
Receive the data transmission interval
sleep
Wake up & Sensing & Random Backoff & data send
(a) Smart Plug
(b) BLE node

Figure 1 Low-energy scheduling algorithm

The BLE node operations are as follows. A BLE node periodically broadcasts its own information to inform the surrounding SmartPlugs. At the same time, the BLE node collects and stores the received information around the SmartPlugs. The BLE node receives the set transmission interval information from the SmartPlug.

The BLE node transmits the data after wakes up and switches into sleep mode, according to the set information. At the same time, after wakes up, a BLE node performs the channel hopping to avoiding collision of data transmission. After that, data is flooding after random-backoff to avoiding collision of data transmission with other nodes on the same channel.

3.2 Frequency hopping of BLE node and data collision

This section examines the frequency hopping and data collision cases. Frequency hopping is a technique for preventing data collision by transmitting data using different channels. If one BLE node is already transmitting data, the other nodes can detect the transmitting data. So, the BLE node performs frequency hopping to avoiding the collision and transmit data using another channel. Channel hopping is done 1,600 times per second, the time for data transmission in one channel is 625 microseconds (1,000,000/1,600 = 625 microseconds).

Figure 2 shows the cases where two nodes can collide when transmitting data. The data transmission starts time of Node 1 denote by s1, and end time denote by e1. Node 2 is also denoted by s2 and e2. In this case, it means the section where the part indicated in red in the picture collides. Of course, if a BLE node knows that another node is already transmitting, performs the channel hopping to another channel, and transmits data. However, it still occurs collisions for the reason such as a hidden problem.

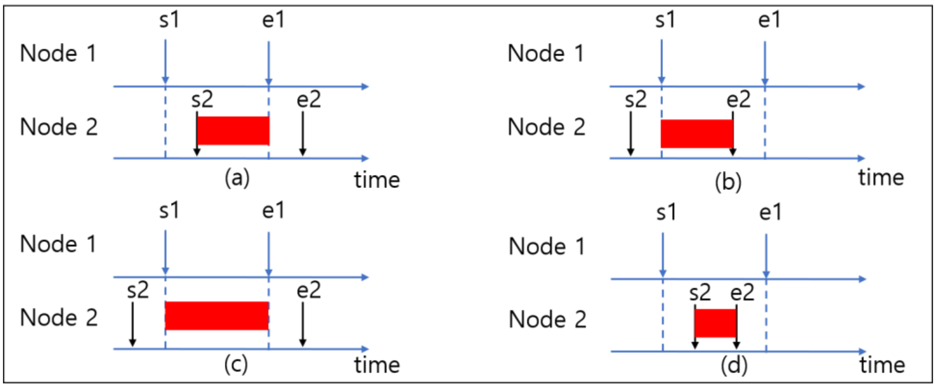

Figure 2 Collision cases of the Node 1 and Node 2

3.3 Collision avoidance method 'random-backoff' and simulation assumptions

This section examines a random-backoff, a very simple method for collision avoidance, and the simulation assumptions. The BLE node uses the frequency hopping. If a BLE node transmits data, another BLE node will not transmit data because one can sense the frequency occupied. Thus, the BLE node changes the frequency, checks the frequency occupied or not, and transmits the data. However, if two BLE nodes try to transmit data at the same time and the same frequency, the collisions can occur. As the number of nodes increases, the collision increases; therefore, this paper proposes a random-backoff. The data transmission attempt time of the BLE node is different after the BLE node performs frequency hopping.

BLE uses 40 channels. BLE performs the frequency hopping 1,600 times per second, so an average of 40 frequency hopping is performed per with one channel per second. Broadcasting channels are 3 of 40 channels and broadcast the data. There are some things to consider for simulation. According to the above calculation, the time available for data transmission through one broadcasting channel is 625 microseconds. However, there is some more time required such as frequency hopping, time to attempt to transmit data, and frequency channel sensing. Therefore, in consideration of the convenience of simulation, $\pm\alpha$ time is required in addition to the data transmission time of 625 microseconds. Also, if each node transmits the data using the different channels, it can avoid the collision. And if each node has a different start time for data transmission after frequency hopping, it can avoid the collision (random-backoff).

4. Simulation results

Figures 3, 4, 5 shows the results of the simulation, and table 1, 2, 3 shows the average of each simulation results. Basically, each node has a different transmission interval for collision avoidance when the simulation. In the once simulation, transmission attempts are 1,000 times, however, some nodes may less 1,000 times because of the transmission interval. In the simulation, increase the number of nodes from 2 to 4, and shows the results of the transmission success ratio with and without random-backoff. The simulation of results shows when the data size is 1~31 bytes and 32~255 bytes, and random-backoff is effective when 1~31 bytes length. In the case of 32 to 255 bytes, random-backoff is not effective, so additional research is needed.

Table 1 shows the different transmission success ratio of node 1 and node 2 applying for random-backoff. Because the transmission interval is different. Node 1 has a short interval, so node 1 has more chances to attempts to transmit the data. As shown in Table 1, the transmission success ratio of node 1 is 94.8% when the data length of node 1 is 1 ~ 31 bytes and apply random-backoff. However, the transmission success ratio of node 1 is 81.4% when the data length of node 1 is 32~255 bytes and applied random-backoff. On the other hand, node 2 has a transmission success ratio is 89.6% when the data length is 1~31 bytes and random-backoff is applied. However, the transmission success rate is 62.8% when the data length is 32~255 bytes and random-backoff is applied. The simulation conducts under the same condition, but it shows that the data length and the number of transmission attempts has an effect. As shown in table 1, in the simulation results show the transmission success ratio is similar regardless of the data length when random-backoff is not applied.

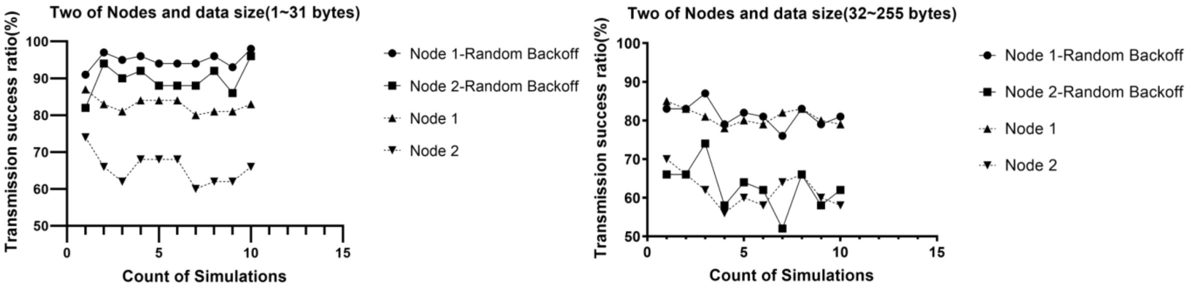

Figure 3 Results of simulations – 2 Nodes

Table1 Average of transmission success ratio (%) – 2 Nodes

	Random-Backoff	***Data size (1 ~ 31 bytes)***	***Data size (32 ~ 255 bytes)***
Node 1	Apply	94.8	81.4
	-	83	81
Node 2	Apply	89.6	62.8
	-	66	62

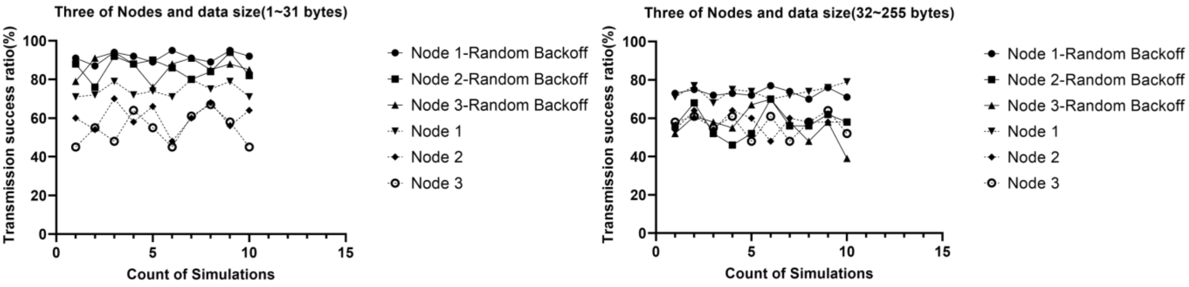

Figure 4 Results of simulations – 3 Nodes

Table2 Average of transmission success ratio (%) – 3 Nodes

	Random-Backoff	***Data size (1 ~ 31 bytes)***	***Data size (32 ~ 255 bytes)***
Node 1	Apply	91.5	73.3
	-	74	74
Node 2	Apply	86	57.6
	-	60	58
Node 3	Apply	86.4	56.4
	-	54	56

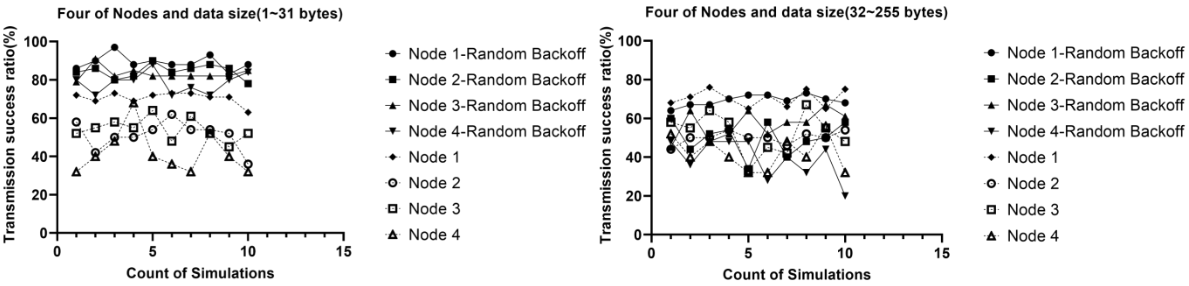

Figure 5 Results of simulations – 4 Nodes

Table3 Average of transmission success ratio (%) – 4 Nodes

	Random-Backoff	*Data size (1 ~ 31 bytes)*	*Data size (32 ~ 255 bytes)*
Node 1	Apply	89.1	69.2
	-	71	70
Node 2	Apply	84.4	49.8
	-	51	50
Node 3	Apply	83	56.7
	-	54	52
Node 4	Apply	78.4	39.2
	-	42	42

It also shows the number of transmission attempts has and effect. In the simulation, the number of nodes is increased to 3 and 4, and the transmission success ratio results are in Figure 4 and Figure 5. And the average transmission success ratio results are summarized in Table 2 and Table 3. As the number of nodes increases, the wireless environment becomes congested, and the transmission success ratio decreases. Also, as the number of transmission attempts decreases, in this simulation results show the transmission success ratio seems to decrease. The simulation results of the number of nodes, 3 and 4 are similar to the number of nodes, 2. So, the rest simulation results are omitted.

5. Conclusions

This paper proposes a BLE mesh network scheduling algorithm using SmartPlug, as well as a random-backoff, which is a very simple method for collision avoidance. In this study, a SmartPlug receives information from BLE nodes around a SmartPlug and schedules the data for the BLE node transmission interval. A SmartPlug can schedule the transmission timing of other BLE nodes for collision avoidance. Also, a random-backoff, which is a straightforward method for collision avoidance, is proposed in this study. Based on the simulation, random-backoff for collision avoidance is effective in transmitting data of 1 to 31 bytes better than 32 to 255 bytes of data length.

6. Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2018-0-01848, Development of a Smart plug and control system based on Wireless convergence for smart factory), (No. 2019-0-0107, Development of hull outfitting inspection system using 3D design drawings based on coexistent reality).

7. References

1. Bluetooth SIG. Introducing Bluetooth Mesh Networking [Internet]. 2017 [July 2017; cited 2019 Aug 7]. Available from: https://www.bluetooth.com/blog/introducing-bluetooth-mesh-networking/
2. Patti G, Leonardi L, Bello L L., A Bluetooth Low Energy real-time protocol for Industrial Wireless mesh Networks. In: Proceeding of the 42nd Annual Conference of the IEEE Industrial Electronics Society; 2016 Oct 23-26; Florence, Italy. New York: IEEE; 2016 [cited 2019 Aug 14]. p. 13-26. Available from: https://ieeexplore.ieee.org/abstract/document/7793093.
3. Bluetooth SIG. 5.1 - Core Specification [Internet]. 2016 [revision 5.1 Jan 2019; cited 2019 Aug 7]. Available from: https://www.bluetooth.com/specifications/bluetooth-core-specification
4. Bluetooth SIG. 1.0.1 – Mesh Profile Specification [Internet]. 2019 [revision 1.01 Jan 2019; cited 2019 Aug 7]. Available from: https://www.bluetooth.com/specifications/mesh-specifications/
5. Bluetooth SIG. 4.2 – Core Specification [Internet]. 2014 [revision 4.2 Dec 2014; cited 2019 Aug 7]. Available from: https://www.bluetooth.com/specifications/archived-specifications/
6. Jing-Ho Chen, Ya-Shu Chen, Yu-Lin Jiang, Energy-Efficient Scheduling for Multiple Latency-Sensitive Bluetooth Low Energy Nodes, IEEE Sensors Journal, 2018 January;18(2): 849-59.
7. Myoung Jin Kim, An Analysis of Packet Collision Probability due to Inter-piconet Interference in the Bluetooth Low Energy Networks, Journal of the Institute of Electronics and Information Engineers, 2018 August; 54(8):1123-31.
8. Murillo Y, Chiumento A, Reynders B, Pollin S. SDN On BLE: Controlling Resource Constrained Mesh Networks. In: ICC2019 - 2019 IEEE International Conference on Communications (ICC); 2019 July 20-24; Shanghai, China. New York: IEEE; 2019 [cited 2019 Aug 0]. p. 1-7. Available from: https://ieeexplore.ieee.org/document/8761066

9. Yaakop M B, Malik I A A, Suboh Z b, Ramli A F, Abu M A. Bluetooth 5.0 throughput comparison for internet of thing usability a survey. In: 2017 International Conference on Engineering Technology and Technopreneurship(ICE2T); 2017 Sept 18-20; Kuala Lumpur, Malaysia. New York: IEEE; 2017 [cited 2019 Aug 1]. p. 1-6. Available from: https://ieeexplore.ieee.org/abstract/document/8215995
10. Imran Ashraf, Athanasios Gkelias, Konstantinos Voulgaris, Mischa Dohler, Aghvami A H. Co-existence of CSMA/CA and Bluetooth. In: 2006 IEEE International Conference on Communications; 2006 June 11-15; Istanbul, Turkey. New York: IEEE; 2006 [cited 2019 Aug 4]. p. 5522-7. Available from: https://ieeexplore.ieee.org/document/4024941
11. Hsu ACC, Wei DSL, Kuo WCCJ. Coexistence Wi-Fi MAC Design for Mitigating Interference Caused by Collocated Bluetooth. 2015 Nov;64(2):342-352.
12. Kondo T, Fujita H, Yoshida M, Saito T. Technology for WIFI/Bluetooth and WiMax Coexistence. 2010 Jan;46(1):72-8.