Wireless Mesh Networking with XBee

Vachirapol Mayalarp, Narisorn Limpaswadpaisarn, Thanachai Poombansao, and Somsak Kittipiyakul
School of Information, Computer, and Communication Technology
Sirindhorn International Institute of Technology, Thammasat University, Pathumthani, Thailand
tba_taker@msn.com, narishardcore@hotmail.com, pondtanachai@hotmail.com, somsak@siit.tu.ac.th

Abstract—**We perform a performance evaluation of ZigBee mesh networks based on XBee modules, in terms of packet delay and received signal strength (RSSI). We evaluate the performance for both line-of-sight (LOS) and non-line-of-sight (NLOS) environments in buildings and a parking lot. The results of our study are useful for building wireless sensor networks and communication networks of mobile robots where the communication distances between nodes are greater than the single-hop range and hence communication between nodes requires multi-hop transmissions.**

I. Introduction

Nowadays, many wireless communication networks are being developed and deployed rapidly, including some of the Wireless Personal Area Networks (WPANs) which are the wireless networks for short ranges. The IEEE categorizes WPANs into two standards: IEEE 802.15.3a for high-speed WPAN and IEEE 802.15.4 or Zigbee for low-speed WPAN. In this paper, we consider a performance evaluation of Zigbee networks.

Zigbee is a standard that defines a set of communication protocols for low-data-rate short-range wireless networking. Zigbee-based wireless devices operate in 868 MHz, 915 MHz or 2.4GHz frequency bands. The maximum data rate is 250 kbps. Zigbee uses DSSS (Direct Sequence Spread Spectrum) which divides the 2.402 to 2.480 GHz spectrum into 16 channels or 10 channels in the 915 MHz spectrum and 1 channel in the European 868 MHz spectrum [1]. The multiple access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) similar to that used in IEEE 802.11. Zigbee is targeted for low-data-rate, low-cost, and battery-powered applications [2]. In many Zigbee applications, the total time that the wireless device is engaged in any type of activity is very limited; the device spends most of its time in a power-saving mode, called ”sleep mode”. As a result, Zigbee-enabled devices are capable of being operational for several years before their batteries need to be replaced [3].

In term of its networking capability, ZigBee provides several types of network topologies ranging from point-to-point, star, to mesh topologies. Mesh networks are suitable for dynamic and large-area networks where nodes are far away from each other and communication between them require multi-hop transmissions and the nodes are dynamic (e.g., mobile) or there are high interference levels in the communication links so that nodes may be considered ”active” or ”inactive” from the network periodically. There are many interesting applications requiring mesh networks, for example, in building and construction automation system [4], [5], in logistics and grain storage [6], [7], in swarm of robots [8], and in vision-based wireless sensor networks [9].

Given such various applications with wireless mesh networking, we are interested in building mesh networks using XBee modules which are ZigBee-compliant chips manufactured by Digi International, Inc. Our initial interest in Xbee is because XBee modules are inexpensive and widely available in Thailand. We are interested in building Xbee-based wireless mesh networks and studying the performance of the networks in term of delay and received signal strengths in the indoor and outdoor environment.

The rest of the paper is organized as follows. In Section II, we give information related to wireless mesh networking with ZigBee/Xbee. Specifically, Section II discusses the types of devices and mesh topology. The experimental results are shown in Section III. Section IV summarizes the paper and discusses some future extension.

II. ZigBee Network Formations

Zigbee defines three different device types, which are coordinator, router and end device [10].

- Coordinator: Start a new PAN by selecting the channel and PAN ID, allow routers and end devices to join the PAN, transmit and receive RF data transmission, and route the data through the mesh network.

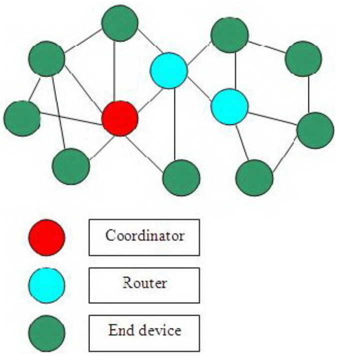

Figure 1. Mesh network

- Router - Transmit and receive RF data transmission, and route data packets through the network.
- End Device - Cannot assist in routing the data transmission but transmit or receive RF data transmission, and intended to be battery-powered devices.

A. *Mesh Topology*

In general, Zigbee mesh topology consists of a coordinator and a set of routers and end devices [10]. A router can be linked to one or more routers and end devices. The communication rules of mesh topology are flexible because the routers that are located within range of each other can communicate directly. An advantage of mesh network is that there is likely another alternative route in case an existing link fails. Hence, this type of network topology is very reliable. Also this concept is applicable to wireless networks, wired networks, and software interaction. The example of mesh topology network is shown in Figure 1.

B. *Building XBee Networks*

Building an XBee network is done automatically by Xbee devices. The Coordinator starts a ZigBee network by scanning for existing PANs in that area and for available channels. It then selects an unused PAN ID and a channel. From now on, a router or an end device can join a nearby Coordinator or another router that has already joined the PAN. If the router or end device is not being a part of the network, it performs a PAN scan of the SC channels and looks for the coordinator or router operating with the valid PAN ID. The router or end device scans the SC channel until the valid device is discovered. Then, it sends an association request frame to associate device and receives an association response to verify the allowance into the network. The joining router and end device then send a broadcast discovery frame to discover 64 bit address of the Coordinator. Each router and Coordinator parents can allow up to 8 end device children to join their own device.

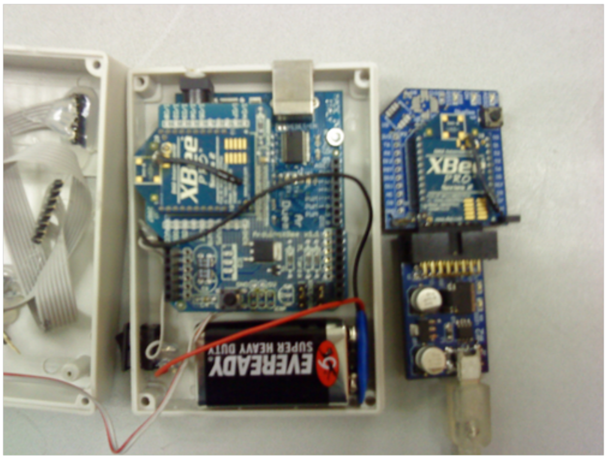

Figure 2. Xbee with Arduino (left) and BlueBee dongle (right)

III. Experimental Study

In our experimental study, we build single-hop and multi-hop Xbee networks with several XBee wireless nodes where each node (shown in Figure 2) consists of an Xbee module and an Arduino microcontroller. The Arduino consists of a simple open hardware design for a single-board microcontroller with embedded I/O support and a standard programming language [11]. C/C++ language is the main programming language of the Arduino. We program the Arduino using Processing [12].

Communication with XBee modules is done either via Arduino or via a USB dongle which is connected to a computer. The XBee module interfaces to a host device through a logic-level asynchronous serial port [10]. The BlueBee dongle shown in the right of Figure 2 allows a serial connection from a computer to XBee via USB port. Using BlueBee dongle together with the X-CTU program provided by Digi [13], we can upgrade firmwares, change device types, configure and monitor system parameters, and send and receive data over XBee modules. Note that the Arduino with XBee shield can also be used as a USB dongle as well.

For a successful deployment of wireless multihop networks, we need to evaluate some basic performance parameters such as radio performance (e.g., received signal strength, coverage ranges and link failure probability), packet delays, and throughputs. In paper, we focus on the measurement of received signal strength (RSSI) and delay performances.

A. RSSI Measurement

RSSI (Received Signal Strength Indicator) is the signal level (in -dBm) of last good packet received. There are two ways to read RSSI value: 1) RSSI value is encoded into pulse-width modulated signal available at Pin 6 of the XBee module, and 2) RSSI value is read via an API command. The RSSI value reported by XBee Pro module is between -36 to -100 dBm while that of a standard Xbee module is between -23 to -92 dBm. However, the XBee manual says that the reported value is accurate between -40 dBm and the sensitivity of Xbee module's receiver [14]. However, we did not verify these numbers in our test.

1) RSSI Measurement in a Parking Lot: In our experiment for RSSI measurement, we connect two XBee Pro modules (i.e., one node is a Coordinator and the other is a Router/End device) and then vary the distance between them to measure the relationship between RSSI values and distances. We make RSSI measurements in line-of-sight (LOS) and non-line-of-sight (NLOS) settings in a parking lot and in a building.

The first results are measured from an SIIT parking lot. First, for the LOS setting we put two Xbee nodes into the vacant area that they can directly ”see” each other. Second, for the NLOS setting we put the two nodes in a parking lot which has some obstructions from cars. The results of the RSSI measurement at various distances are shown in Figure 3. In Figure 3, the x-axis is the distance measured in meters and y-axis is the RSSI value measured in dBm. As expected the measured RSSI values for LOS and NLOS setting decrease with the distance and the NLOS RSSI values are much lower than the LOS values due to obstructions.

2) RSSI Measurement in a Library: Another location that we measure the RSSI is a library at SIIT. We perform on RSSI measurements in NLOS setting, where we fix the position of the Xbee Coordinator node in one room and move the Router/End device node away from the Coordinator in to the room next to the Coordinator's room and measured the RSSI. We did this for three more times in three other different rooms. The RSSI measurement result is shown in Figure 4.

From Figure 4, we can see that the RSSI decreases rapidly with distance for the first three rooms but slightly decreases in the fourth room. The RSSI decays linearly with distance (this result is similar to the results from the parking lot above). The cause of the slight decrease in the fourth room is still unknown to us.

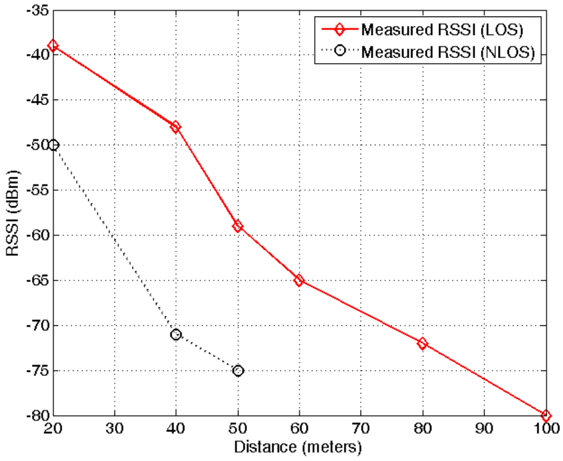

Figure 3. Measured RSSI values versus distance, in LOS and NLOS settings, in a parking lot.

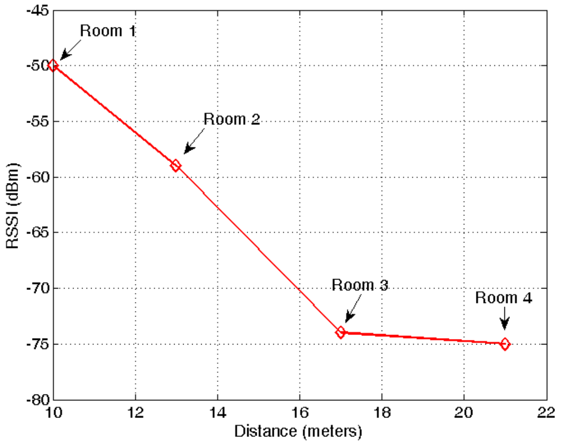

Figure 4. Measured RSSI versus distance from the transmitting node to the receiving node located in four rooms of the library.

B. Packet Delay Measurement

To measure packet delay in XBee networks, we program the Arduino to send the ”Hello World” sentence, which has the size of 10 Bytes. We send the sentence in every 5 seconds to the Coordinator and wait until the Coordinator responses back with an acknowledgment packet (ACK) back to the source. The packet delay in our experiments is defined to be the duration between sending a packet until its acknowledgment packet has been received by the source. This may be different from the standard definition of packet delay where a one way delay from the source to destination is considered. Our delay is defined as the round-trip time (RTT) until an

ACK is received.

Note that when a packet is transmitted from a source node to a destination node (possibly via multiple hops), a network-level acknowledgment is transmitted back from the destination node across the established route to the source node. This acknowledgment packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgment is not received, the source node will re-transmit the data [10].

1) Delay Measurement in Point-to-Point Link: We first perform a measurement of the packet delay in a point-to-point link. We connect an Xbee module that act as the Router/End device to the Arduino board which sends "Hello World" packets to the Coordinator, receives ACK packets from the Coordinator, and calculate the delay times. We observe the results of the delay time around 64 to 67 ms while the two Xbee modules is still able to communicate to each other until the nodes are out of range.

2) Delay Measurement in Multi-Hop Networks: In the second experiment, the delay measurement is done in a dormitory building with three Xbee modules. We position the three modules as shown in Figure 5. Nodes A, B, and C have the following duties:

A: ZNet 2.5 Coordinator
B: ZNet 2.5 Router/End device
C: ZNet 2.5 Router/End device

We fix the location of Node A, the Coordinator, and have Node C join the network. Then, Node C sends a data packet to A so that we can check the RSSI at Node A. Then we gradually move Node C further away from Node A until the link was disconnected. Right at the disconnecting distance, the RSSI value of RSSI at Node A is about -79 dBm while Node C is located as shown in Figure 5. Next, we add Node B into the network and located as shown in Figure 5. With the relay help at Node B, Node C can once again transmit data packets to Node A. At this time, the RSSI at Node A is about -63 dBm. But when we move node C closer to node B, the value of RSSI changes into -79 dBm while node B is still in the network. This means that node C can communicate with node A directly without node B's relaying.

To measure packet delay times in networks with more than two hops, we use the Arduino to send "Hello World" to the Coordinator (node A). We repeat the previous experiment to create a network with three hops and four nodes, as shown in Figure 6, where node D is the source node and node A is the destination node. Nodes A, B, C, and D have the following duties:

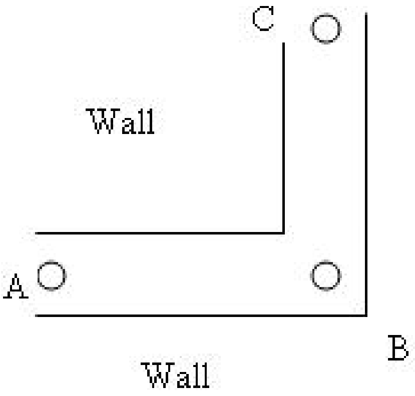

Figure 5. Nodes' positions for delay measurement in a two-hop setting

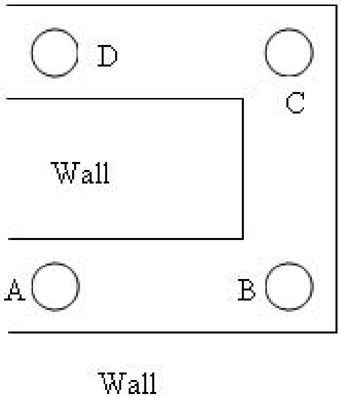

Figure 6. Nodes' position for three-hop testing

A: ZNet 2.5 Coordinator
B: ZNet 2.5 Router/End device
C: ZNet 2.5 Router/End device
D: ZNet 2.5 Router/End device

The delay measurement results for multiple hops are shown in Figure 7. The packet delays for two to four hops are almost similar to that of the single-hop setting, i.e., the delays are about 64 to 67 ms. Hence, the number of hops does not have much effect on the packet delay which is defined based on the ACK reception. Normally we would have expected that the delay shall increase significantly with the number of hops because of the extra processing delay at the router and transmission delay due to an additional hop. However, these delays are small. For example, given that the link speed is 250 kbps, the transmission delay is about $10 * 8/250 = 0.32$ msec, ignoring the overhead in the MAC and network

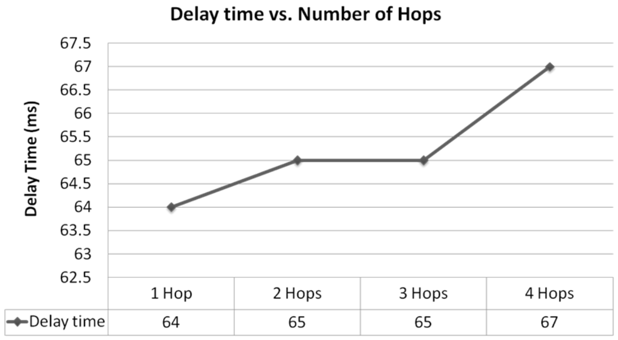

Figure 7. Number of Hops versus delay time

layers. This transmission delay is much smaller than the total delay of 64 msec. Given the results in Figure 7, we can also conclude that the delay due to packet forwarding at each intermediate router is negligible as well.

IV. Conclusions and Future Work

This paper reports our initial study of Xbee-based wireless mesh networks. We evaluate two main performance parameters: the received signal strength (RSSI) and the round-trip packet delay. We measure a relationship between RSSI and distance for a point-to-point link in both line-of-sight (LOS) and non-line-of-sight (NLOS) environment. Next, we measure the packet delay in both star (single-hop) and mesh (multi-hop). The delay that we measure is the loopback or RTT delay for the ACK from the application sublayer. These delays vary little with the more number of hops. Hence, we could conclude that the main part of the delay is due to getting the ACK from the application layer.

Future extensions of this work include an automatic monitoring of RSSI in real time systems, testing for battery consumption time, network life time, and measuring the parameters (RSSI and packet delay) in larger network settings. We also plan to include the mesh networking capability into the temperature monitoring system being developed in parallel by our colleague [15].

References

[1] *XBee/Xbee Pro OEM RF Modules*, MaxStream, Aug. 2009.

[2] D. Gislason, *Zigbee wireless networking*. San Juan Software, 2008.

[3] ZigBee Alliance. [Online]. Available: http://www.zigbee.org/

[4] J. Raimo, "Wireless mesh controller networks," *ASHRAE Journal*, vol. 48, no. 10, pp. 34–38, 2006.

[5] W.-S. Jang, W. M. Healy, and M. J. Skibniewski, "Wireless sensor networks as part of a web-based building environmental monitoring system," *Automation in Construction*, vol. 17, no. 6, pp. 729–736, Aug. 2008.

[6] L. Ruiz-Garcia, P. Barreiro, and J. Robla, "Performance of ZigBee-based wireless sensor nodes for real-time monitoring of fruit logistics," *Journal of Food Engineering*, vol. 87, no. 3, pp. 405–415, Aug. 2008.

[7] H. Zhoul, X. Chen, X. Liu, and J. Yang, "Applications of Zigbee wireless technology tomeasurement system in grain storage," *Computer and Computing Technologies in Agriculture II, Volume 3*, pp. 2021–2029, 2009.

[8] A. Turgut, H. Aelikkanat, F. Gokce, and E. Sahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, no. 2, pp. 97–120, Dec. 2008.

[9] A. Rowe, D. Goel, and R. Rajkumar, "FireFly Mosaic: A vision-enabled wireless sensor networking system," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium.* IEEE Computer Society, 2007, pp. 459–468.

[10] *Xbee ZNet 2.5/XBee Pro ZNeet 2.5 OEM RF Modules*, v1.x.4x ed., Digi International Inc., Nov. 2008.

[11] Arduino. [Online]. Available: http://arduino.cc/

[12] Processing. [Online]. Available: http://processing.org/

[13] *X-CTU Configuration and Test Utility Software: User's Guide*, Digi International Inc., Aug. 2008.

[14] *XBee/Xbee Pro OEM RF Modules: Product Manual v1.xEx - 802.15.4 Protocol*, Digi International, Inc., Sep. 2009.

[15] V. Boonsawat, J. Ekchamanonta, K. Bumrungkhet, and S. Kittipiyakul, "Xbee wireless sensor networks for temperature monitoring," in *2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010)*, Pattaya, Chonburi, Thailand, 10-12 May 2010.