ZigBee Based Self-Organizing Network

A. Ashok Kumar¹, Ch. Saideep², Y. Lalita Krishna³, P. Srishaka⁴

¹Assistant Professor, ²³ & ⁴B.Tech. IVth Year Students,
Department of Electronics & Communication, MVGR College of Engineering,
Vizianagaram, Andhra Pradesh, India

Abstract — The Internet of Things (IoT) is the network of physical objects—devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity—that enables these objects to collect and exchange data. For the network to operate smoothly all the time, there must be a mechanism to detect failures in any of the nodes in the network topology and automatically assign a new pathway so that the communication does not get affected. This paper Zigbee based Self-Organizing Network investigates a way for any IoT network that uses Zigbee protocol for communication to dynamically assign an alternate path in the mesh network i.e. self-organize, whenever a node through which data is being routed fails.

Keywords — ZigBee, Internet of Things, Self-organizing network, ZigBee coordinator, ZigBee router, ZigBee end device, Wireless sensor networks, Sensor nodes.

I. INTRODUCTION

The collection of sensor nodes systematized into a network is known as WSNs. A sensor is a small device which observes the environment of physical parameters like pressure, temperature, sound, pollutants or vibration. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial, consumer and medical applications, such as industrial process monitoring and control, patient health monitoring, and so on. Wireless sensor networks are a subset of Internet of Things (IoT).

For example, consider a daily life situation of a patient who is staying at his home but for whom it is important that his physician monitor his heart rate and blood pressure continuously. In this system, a mesh network can be used to collect data from various sensors connected to the patient. Whenever a node fails to forward the data from patient to physician, a situation might arise where the physician could not identify the problem is with the network or the patient. In such cases a self-organizing network is required.

A Self-Organizing Network is an automation technology designed to make the planning, configuration, management, optimization and healing of wireless mesh networks simpler and faster. Fig. 1 shows a simple ZigBee mesh network.

Self-organization can be classified into four aspects as follows.

- **Self-configuration**: The ability of WSN to automatically and seamlessly configure sensor nodes.
- **Self-healing**: The ability of WSN to automatically detect, diagnose, and repair localized software and hardware problems.
- **Self-optimization**: The ability of WSN to continually seek opportunities to improve their own performance and efficiency.
- **Self-protection**: The ability of WSN to automatically defend themselves against malicious attacks or cascading failures. A WSN should use early warning to anticipate and prevent system wide failures.

![Figure 1: A ZigBee Mesh Network](image-url)
technology for contactless and automated identification. ZigBee-based wireless devices operate in 868MHz, 915 MHz, and 2.4 GHz frequency bands. It is based on physical and medium access control layers defined in IEEE 802.15.4 specifications. The maximum data rate of ZigBee is 250 kilobits per second. ZigBee is targeted mainly for battery-powered applications where low data rate, low cost, and long battery life are main requirements. As a result, ZigBee enabled devices are capable of being operational for several years before their batteries need to be replaced. ZigBee supports mesh networks and has self-organizing capability hence suitable for applications like mentioned above.

II. A SIMPLE SELF-ORGANIZING NETWORK

A. ZigBee Basics

The ZigBee protocol defines three types of nodes: Coordinator (C), Routers (R), and End Devices (E) as shown in the sample mesh network in Fig. 1.

A coordinator establishes the network originally. There is exactly one coordinator in each network. Routers act as intermediate nodes, relaying data from other devices. End devices, usually sensors, can be low-power devices. They have sufficient functionality to talk to their parent devices, either the coordinator or a router, but cannot relay data from other devices.

This simple test mesh network consists of four nodes with one coordinator, two routers and an end device as shown in Fig. 2, where ZC represents ‘ZigBee Coordinator’, ZR represents ‘ZigBee Router’ and ZED represents ‘ZigBee End Device’.

An infrared proximity sensor is used at the end device whose feed is continuously provided to the coordinator. An LED is used at the coordinator which glows whenever an obstacle is detected at the end device by the sensor. As can be seen in Fig. 2, there are two possible paths for the end device to reach the coordinator viz. A-B-D and A-C-D. When the network is formed a path is chosen depending on several factors or link cost like number of hops, signal strength etc. Let the path be assumed as A-C-D. Now it can be experimentally verified that if the router at node C is removed or powered down, a new path A-B-D is automatically discovered thus proving that this simple ZigBee mesh network is self-organizing.

There are several route discovery algorithms used in ZigBee protocol that include ad-hoc on-demand distance vector routing (AODV) algorithm, source routing and many-to-one routing. The XBee modules used can be set to use either of these routing algorithms. AODV route discovery protocol is used for the purpose of demonstration in this paper.

B. AODV Routing Algorithm

In AODV routing, whenever a sending node needs a path discovery, it issues route request command (RREQ) to all its neighbouring nodes. These nodes forward the request to their neighbouring nodes and this goes on till the destination node is reached. Then the destination node generates a route reply (RREP) command which will be reverted to the source node. The source node stores only the next hop information for each node in its own routing table. The AODV route discovery process for a general mesh network is shown in Fig. 3. Here A is the source node and J is the intended destination node.
C. Hardware Setup

1) XBee Wireless Modules: XBee wireless modules which are based on ZigBee protocol are used at all the nodes of this simple mesh network[9]. A single XBee Series 2 module can be configured as a ZigBee coordinator, router or end device. XBee modules can work independently once they are configured i.e. standalone or they can be interfaced to any external microcontroller. The software tool XCTU is used to configure the XBee modules. Once the network is formed, it can also be used to view the network topology as will be described later.

2) Arduino Uno: It is an open-source physical computing platform based on a simple microcontroller board and a development environment (IDE) for writing software for the board. The Arduino Uno is a microcontroller board based on the ATMega328 microcontroller. In this test network, Arduino is used at the coordinator and end device. The XBee module is interfaced to Arduino[9] using an XBee shield. The shield can be directly mounted on to the Arduino board. Routers doesn’t require Arduino for this demonstration, they can work standalone.

Fig. 4 shows the ZigBee end device (ZED) having an Arduino interfaced to XBee module and infrared proximity sensor.

![Figure 4: ZigBee End Device (ZED)](image)

III. RESULT AND DISCUSSION

The XBee communicates with the Arduino using the TX and RX lines. In order to transmit any data from Arduino over XBee or to receive data from XBee to Arduino, the default Serial library[11] can be used that is provided by Arduino. As an example, the code below can be used to send a character serially from Arduino to XBee for transmission.

```java
void setup(){
  Serial.begin(9600);
  //Baud rate set at 9600 bps
}
void loop(){
  Serial.write ('A'); //Send character A
}
```

Similarly, “Serial.read” method can be used to read data received at other end by Arduino serially as shown below.

```java
void setup(){
  Serial.begin(9600);
  //Baud rate set at 9600 bps
}
void loop(){
```
In this demonstration, the Arduino at the end device is interfaced to an infrared proximity sensor. Its raw analog values are read by the Arduino and given as input to the XBee which is also interfaced to this Arduino. The XBee transmits these values to the coordinator via any of the two routers. The XBee at the coordinator receives the values and sends them to the Arduino interfaced to the XBee. The Arduino is programmed to control the LED depending on the values received.

The network once formed can be viewed in XCTU [9] as shown in Fig. 8. It can be seen that the end device communicates with the coordinator (C) through the ZigBee router (R) addressed as 2CDA. This unique 16-bit address, also called as short address, is assigned to all the nodes by the coordinator. The unique 64-bit MAC address as can be seen in Fig. 8 can also be used to refer the nodes.

![Figure 8: Network topology with 4 nodes as seen in XCTU](image)

Now the router with short address 2CDA, to which the end device is connected is removed and the coordinator stops receiving from the end device. It can be observed that after a brief delay, the end device reconnects with the coordinator through the other router addressed as 5A35 which can be seen in XCTU [9] window shown in Fig. 9. It can also be seen that the earlier router addressed as 2CDA is shown in red as it is no longer part of this network.

![Figure 9: Network topology after a router is disconnected. A new path between E and C can be seen](image)

A new path to the coordinator is automatically discovered thus demonstrating the self-organizing capability of wireless mesh networks that use ZigBee protocol.

The patient monitoring system explained in the beginning is one such example where a network outage can have drastic consequences. Self-organizing networks are also useful in IoT applications like home automation, industrial automation, building security etc. for forming a reliable and robust communication systems without the need of urgent manual intervention in case of any node failure. ZigBee protocol intended for low cost, low power, low data rate applications have self-organizing capability and hence is suitable for such applications. Other wireless protocols that include Bluetooth, Wi-Fi etc. doesn’t have self-organizing property.

This paper demonstrates how ZigBee wireless protocol can be used to construct robust self-organizing networks. There are some more algorithms to detect and correct a node failure like Distributed Cut Detection (DCD) [13] but they require higher processing power than ZigBee and hence costlier to implement. Moreover, ZigBee certified devices enable interoperability among devices from different manufacturers thus making them a simple plug-and-play solution without the need to customize the algorithm for different manufacturer products. Also, adopting a common specification like ZigBee reduces the burden on manufacturers and they can focus on innovative applications in IoT instead of solving practical problems like these.

IV. CONCLUSION

As can be seen from this demonstration, the coordinator continues responding to the sensor input and a node failure did not cripple the entire network apart from a short communication failure which will be about 2 – 5 seconds.
A simple application of obstacle detection is implemented to demonstrate the self-organizing property that is incorporated in ZigBee protocol.

V. REFERENCES