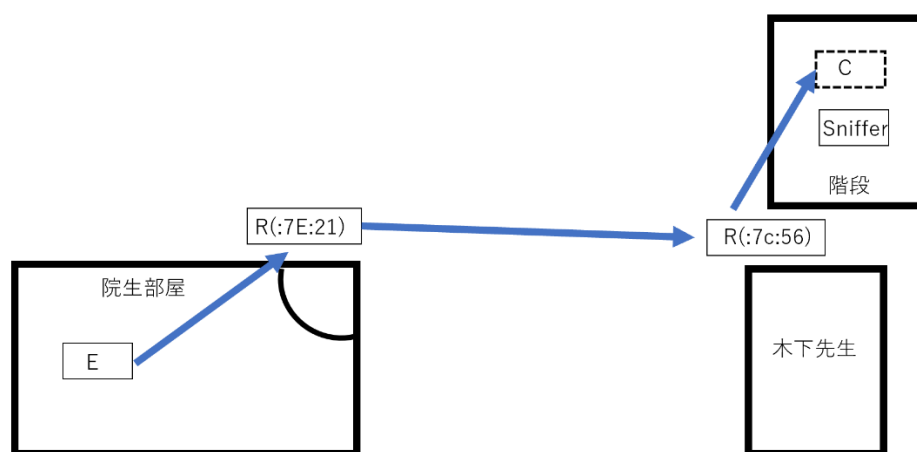


0607

【1】以前アルベルト先生に指摘していただいたネットワークの信頼性についての予備実験について、以下はアルベルト先生のステップを抜粋。

- a) マルチホップ・ネットワークを設定し、マルチホップ方式で動作していることを確認する（コーディネータとエンド・デバイスの間に少なくとも3～4本のリンクがある）。
- b) センサーから実データを作成し、送信先に送信する（複数回のテスト）。送信されたデータは100%宛先に届いているか？そうでない場合は、そうでない理由を見つける。
- c) ノードの1つが削除された場合、ネットワークは回復できるか？



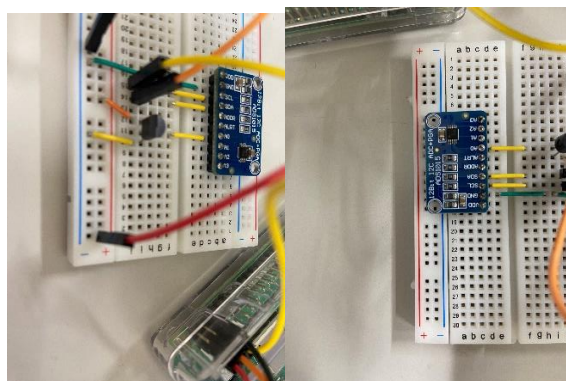
Coordinator（受信側）1 台

Router 2 台

（MAC アドレス 00:1b:c5:01:22:03:7e:21 と 00:1b:c5:01:22:03:7c:56）

Enddevice（送信側）1 台

Enddevice では温度センサとラズパイに接続されている。



そして Sniffer でデータの流れを調査した。

## [実験の流れと結果]

1. Coordinator を起動.
2. Router (:7c:56)をコンセントに接続する.
3. Router (:7e:21)をコンセントに接続する.
4. Enddevice をラズパイに接続し, 温度センサでデータを取得していく.

以下はこの時点での, Coordinator の取得状況である.

a0. a1...は, 受信した順である.

0x1465 の shortMAC アドレスは送信元である Enddevice である.

```
Problems Console Properties Router EndDevice Coordinator 100En
Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)
a0. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a1. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a2. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a3. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a4. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a5. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
a6. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a7. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
a8. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
a9. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
aa. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
ab. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
ac. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
ad. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
ae. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
af. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
b0. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
b1. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
b2. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
b3. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
b4. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
b5. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
b6. [MAC: 0x1465][Data16: 0x00dbf97e6abc74eb3f000000000000]
b7. [MAC: 0x1465][Data16: 0x003108ac1c5a64eb3f000000000000]
```

実行結果より, Enddevice から送信されたデータは 100%漏れなく, 送信先である Coordinator に届いている.

この時点では, 接続図のように Coordinator の一つ前の Router は :7c:56 であった.

```
1 1 IEEE 802.15.4 Sniffer Protocol
1 1 IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x3183
1 1   Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, PAN ID Compr
1 1   Sequence Number: 24
1 1   Destination PAN: 0x469b
1 1   Destination: 0x0000
1 1   Source: 0x3183
1 1   [Extended Source: IEEERegi_01:22:03:7c:56 (00:1b:c5:01:22:03:7c:56)]
1 1   [Origin: 4]
1 1   FCS: 0xd5c1 (Correct)
1 1   ZigBee Network Layer Data, Dst: 0x0000, Src: 0x1465
1 1
```

5. しばらくして, Router (:7c:56)をコンセントから抜く.  
以下が, この時点での Coordinator の受信状況である.

```
Problems Console Properties Router EndDevice Coordinator 100Endd
Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)
1d. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
1e. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
1f. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
20. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
21. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
22. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
23. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
APP: vCheckStackEvent: ZPS_EVENT_NWK_STATUS_INDICATION
24. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
APP: vCheckStackEvent: ZPS_EVENT_NWK_STATUS_INDICATION
25. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
26. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
27. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
28. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
29. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
2a. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
2b. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
APP: vCheckStackEvent: ZPS_EVENT_NWK_STATUS_INDICATION
2c. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
APP: vCheckStackEvent: ZPS_EVENT_NWK_STATUS_INDICATION
APP: vCheckStackEvent: ZPS_EVENT_NWK_STATUS_INDICATION
2d. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
2e. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
2f. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
```

エラーが表示されていくが, 順番通りに受信できている.  
この時点では, 接続図のように Coordinator の一つ前の Router は :7e:21 であった.

```
IEEE 802.15.4 Data, Dst: 0x3183, Src: 0xf445
  Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, PAN ID C
  Sequence Number: 191
  Destination PAN: 0x469b
  Destination: 0x3183
  Source: 0xf445
  [Extended Source: IEEERegi_01:22:03:7e:21 (00:1b:c5:01:22:03:7e:21)]
  [Origin: 16]
  FCS: 0x4b46 (Correct)
  ZigBee Network Layer Data, Dst: 0x0000, Src: 0x1465
```

6. そのまま, Router (:7e:21)をコンセントから抜く.  
この時点で, Coordinator で受信できなくなり, 完全に停止した.
7. Router (:7c:56)と Router (:7e:21)をコンセントに接続する  
以下がこの時点での, Coordinator の受信状況である.  
止まっていた受信が, 再び開始された.  
この時点では, 接続図のように Coordinator の一つ前の Router は :7c:56 であった.

```
IEEE 802.15.4 Data, Dst: 0x0000, Src: 0x3183
  Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, PAN ID C
  Sequence Number: 19
  Destination PAN: 0x469b
  Destination: 0x0000
  Source: 0x3183
  [Extended Source: IEEERegi_01:22:03:7c:56 (00:1b:c5:01:22:03:7c:56)]
  [Origin: 4]
  FCS: 0xd7c9 (Correct)
  ZigBee Network Layer Data, Dst: 0x0000, Src: 0x1465
```

```
Problems Console Properties Router EndDevice Coordinator 100Endc
Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)
8b. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
8c. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
8d. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
8e. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
8f. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
90. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
91. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
92. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
93. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
94. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
95. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
96. [MAC: 0x1465][Data16: 0x01dbf97e6abc74eb3f000000000000]
97. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
98. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
99. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
9a. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
9b. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
9c. [MAC: 0x1465][Data16: 0x018716d9cef753eb3f000000000000]
9d. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
9e. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
9f. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
a0. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
a1. [MAC: 0x1465][Data16: 0x013108ac1c5a64eb3f000000000000]
```

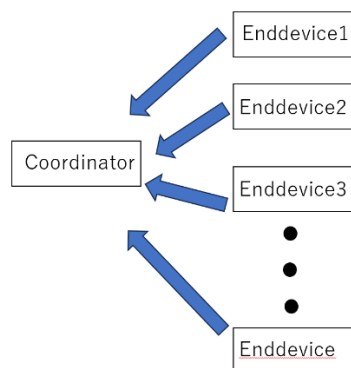
以上が実験の流れと結果である。

もう一度、同じ条件で実験を行ってみても、結果は同じになった。

a)～c) をクリアしており、実験を行った B2 研究室は電波をあまり通さない環境であるため、院生部屋から 3F の階段までデータを送受信できたことで、マルチホップの利点を生かすことができたと言える。

d) ネットワーク内の多くの場所からセンサーデータを同時に送信する。

ラズパイが複数個無いため、作成した文字列送信を行っている。先週に以下のような環境を構築し、Enddevice 3 台の場合、4 台の場合、5 台の場合で実験を行った。



0.001s での結果

[Enddevice 3 台の場合] 漏れなし。

[Enddevice 4 台の場合] 漏れなし。

[Enddevice 5 台の場合]

COM3 : 0～255 で漏れなし。

COM4 : 44～230

COM7 : 121～255

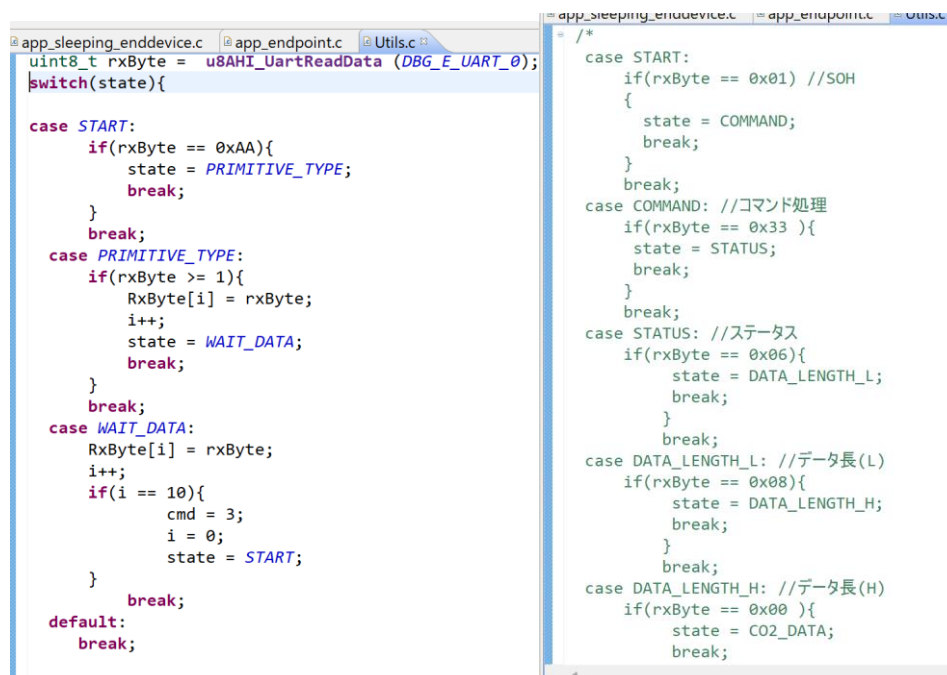
COM9 : 180~255

COM10 : 0~43, 241~255

0.1s, 0.01s の場合は, Enddevice5 台でも漏れなく, Coordinator で受信することができた.

f) センサーの種類や送信するデータ量を変更した場合、プロトコルやアプリケーションは簡単に変更できますか？設計は柔軟であるべきです。

現在は以下のように、センサの送受信はラズパイからの受信データごとに処理を行っている。また、辻先生からいただいた実際の養殖場で使用されているセンサのデータに合わせたコードも作成はしてある。



```
app_sleeping_enddevice.c | app_endpoint.c | Utils.c
uint8_t rxByte = u8AHI_UartReadData(DBG_E_UART_0);
switch(state){
case START:
    if(rxByte == 0xAA){
        state = PRIMITIVE_TYPE;
        break;
    }
    break;
case PRIMITIVE_TYPE:
    if(rxByte >= 1){
        RxByte[i] = rxByte;
        i++;
        state = WAIT_DATA;
        break;
    }
    break;
case WAIT_DATA:
    RxByte[i] = rxByte;
    i++;
    if(i == 10){
        cmd = 3;
        i = 0;
        state = START;
    }
    break;
default:
    break;
}
```

```
/*
case START:
    if(rxByte == 0x01) //SOH
    {
        state = COMMAND;
        break;
    }
    break;
case COMMAND: //コマンド処理
    if(rxByte == 0x33 ){
        state = STATUS;
        break;
    }
    break;
case STATUS: //ステータス
    if(rxByte == 0x06){
        state = DATA_LENGTH_L;
        break;
    }
    break;
case DATA_LENGTH_L: //データ長(L)
    if(rxByte == 0x08){
        state = DATA_LENGTH_H;
        break;
    }
    break;
case DATA_LENGTH_H: //データ長(H)
    if(rxByte == 0x00 ){
        state = CO2_DATA;
        break;
    }
}
```

## 【2】辻先生からのコメント

Zigbee メッシュネットワークの構成

- ・ 正常時と故障時のシミュレーション  
正常時と故障時の通信時の遅延など検討
- ・ 遮蔽物を想定したシミュレーション  
遮蔽物ありとなしの比較検討
- ・ 通信時のスループット、通信経路に関する時間、遅延、損失、電波強度などを比較  
RSSI の使用する.

シミュレーションのシナリオとして、工場の水槽や金属製の棚、扉、冷蔵庫、配管などを想定して、遮蔽物があった場合に、Zigbee の自由なデバイス配置とネットワーク構成で、通信の損失無くデータが送受信できるかを確認する.

1：デバイス数は減らせるが、経路上の Router が故障したときスループットが落ちる.単位時間辺りのデータ数が、正常な通信時と故障時の通信時間を比較してグラフ化する.

Router 2 台を並列にしてメッシュ構成として冗長化する.Router 1 台が故障しても問題なく (Router の切り替えが行われ) データが送れるか確認する. 同じく、スループットを計測する.

2：Router の切り替えに関する遅延が起きていないか確認. スループットを測るのに、Coordinator でデータ到着時の時刻情報を取得する. Enddevice と Coordinator で時刻が同期できていれば、Enddevice からデータ送信時にタイムスタンプを付けて送る.Coordinator の受信時刻との差分から、データ送信時に通信経路にかかった時間が判明する.

【3】実際に水が入った容器(水槽の代わり)を用意し、データの受信状況を確認.  
未実施.