

修士学位論文

牡蠣養殖における育苗を支援する
無線マルチホップネットワーク

Wireless Multi-hop Network to Support Seedling
Cultivation in Oyster Farming

指導教員：ガジェゴス ラモネト アルベルト助教

提出者：山本郷平（6123350461）

令和7年2月14日

徳島大学 大学院創成科学研究科
理工学専攻 知能情報システムコース

牡蠣養殖における育苗を支援する 無線マルチホップネットワーク

山本郷平 (6123350461)

内容梗概

本研究では、水質や温湿度をリアルタイムに管理しながら給餌量を調整する牡蠣養殖場の育苗施設内に各種センサを導入して、自動的に各データを収集するシステムについて検討する。施設内では狭い空間に多くのタンクが設置されているため、作業スペースが限られ無線通信技術が不可欠な状況である。現在、センサによる水質情報のデータ収集のみ実現できているが、水槽内の水質センサと水槽の外にある通信モジュールを有線で接続しているため、設置運用の容易性を考慮してセンサと通信モジュールの一体化を提案する。一体化したセンサと通信モジュールを水中に沈めて、ZigBeeとIEEE802.15.4で水槽の外にあるサーバに送信するが、水中において電波は大幅に減衰するため、実機を用いた予備実験によって水中からの通信可能距離を測定して実現可能性を示した。その上で、中継機の役割をもつ通信モジュールを水中に沈めた通信モジュールとサーバの間に設置することで、水中からの通信モジュールのデータを水槽外のサーバに送信できるようにする無線マルチホップネットワークを提案する。そして、無線マルチホップネットワークを実装して動作確認を行うことで、本研究でのシステムの実用性を示すことができた。

キーワード

ZigBee, 無線マルチホップネットワーク, IEEE802.15.4, 牡蠣養殖

Wireless Multi-hop Network to Support Seedling Cultivation in Oyster Farming

Kyohei Yamamoto

Abstract

In this study, we introduce various sensors into the nursery facilities of oyster farms, which adjust the amount of food fed while managing water quality, temperature and humidity in real time, and consider the system that automatically collects data and supports seedling cultivation. Many tanks are installed in a confined space within the facility, resulting in limited working space. Consequently, wireless communication technology is essential. Currently, only water quality information data can be collected by sensors, but since the water quality sensor in the tank is connected to the communication module outside the tank by wire, we propose integrating the sensor and communication module in consideration of ease of installation and operation. The integrated sensor and communication module are submerged in water and transmit data to a server outside the tank using ZigBee and IEEE802.15.4. However, since radio waves are significantly attenuated underwater, we measured the communication distance from underwater in a preliminary experiment using an actual device and demonstrated the feasibility. We then propose a wireless multi-hop network that allows data from the communication module to be transmitted from underwater to a server outside the tank by installing a communication module that acts as a repeater between the communication module submerged in water and the server. We then implemented the wireless multi-hop network and confirmed its operation, demonstrating the practicality of the system in this study.

Keywords

ZigBee, wireless multi-hop network, IEEE802.15.4, oyster farm

目次

| | |
|---------------------------------|----|
| 第1章 序論 | 1 |
| 第2章 牡蠣養殖の過程 | 3 |
| 第3章 予備実験 | 4 |
| 3.1 センサと通信モジュールの一体化 | 4 |
| 3.2 実験端末 | 4 |
| 3.3 水中での通信品質評価 | 8 |
| 3.4 養殖場でのマルチホップ通信確認 | 16 |
| 第4章 本実験 | 17 |
| 4.1 無線マルチホップネットワークの提案 | 17 |
| 4.2 無線マルチホップネットワークの実装 | 17 |
| 4.3 無線マルチホップネットワークの実験 | 19 |
| 第5章 結論 | 22 |
| 謝辞 | 23 |
| 参考文献 | 24 |

第1章 序論

近年，日本国内における農林水産業では人手不足が深刻な問題になっているため，各業界ではIoT技術の適用が盛んに行われている．例えば，日本の漁業全体の就業者数[1]を見ると，2004年には約23.1万人だった人数が，2022年には約12.3万人に減少している．また新規漁業就業者数[2]も減少傾向にあり，2018年には約1943人が新たに漁業に就業していたのに対し，2019年には約1729人で約1割減少しており，若年層の参入が減少していることが課題となっている．そこで，水産業に関する研究[3]では，日々の給餌量をセンサで記録管理することで，環境と給餌量の相関関係を分析し，給餌量を最適化している．また，農業に関する研究[4]では，作物の近くに設置したセンサとカメラで得たデータを収集し，AIで分析と予測をしている．その結果を制御装置に反映することで自動化と高品質と高収穫を実現している．林業に関する研究[5]では，木材に関するデータを無線で収集し可視化と自動集計を行うことで，人の手で行っていた検木作業を削減している．このように各業界でIoT技術の導入が進められており，人手不足の解決に努めている．

本研究は，IoT技術を活用して牡蠣養殖の作業を効率化して生産性を向上させることを目的とした取り組みとして徳島県内の企業と共同で行われている「スマート牡蠣養殖」[6]というプロジェクトの一環として取り組んでいる．牡蠣の成長や健康は，水温，塩分濃度，酸素濃度といった水質に大きく依存している．これらの環境要因をリアルタイムで監視することで，適切な環境を維持して，生産ロスを最小限に抑えることが可能である．スマート牡蠣養殖では，養殖場に設置されたセンサで水温，塩分濃度，酸素量などのデータを取得してRaspberry Piなどのコンピューティングデバイスで処理する．その後，センサデータはクラウドやローカルサーバに送信され，スマートフォンやPCを通じて直観的に可視化される．このように，スマート牡蠣養殖では従来の手作業をIoT技術で自動化させることで，効率的かつ持続可能な養殖業を実現し，新規の牡蠣養殖業者が参入しやすい環境作りを目指している．

本研究では，母貝の育成から稚貝までを育て上げる施設内に注目している．狭い空間に多くのタンクが設置されているため，作業スペースが限られている．そのため，施設内では無線通信技術が不可欠である．現在，センサによる水質情報のデータ収集のみ実現できている．しかし，水槽内の水質センサと水槽の外にある通信モジュールを有線で接続しているため，牡蠣に接触しないような設置や経年劣化による頻繁な交換などの問題が発生している．この問題を解決する方法として，通信モジュールを防水ケースに収め，浮体に取り

付けて水面に浮かべる方法が考えられる。この方法では通信モジュールから水質センサまでの有線接続が短縮されて、設置の簡便化が図られる。また、牡蠣への接触リスクを低減でき、通信モジュールが水面にあるため電波の信号強度を低下させることなく通信ができるというメリットがある。一方で、浮体部分が水面上で障害物となる可能性や防水ケースや浮体の劣化、さらには水面付近のみの使用に限定されるといった課題も想定される。そこで本研究では、センサと通信モジュールを一体化することを提案する。その際、通信モジュールの電源として小型バッテリーが必要となることに注意する。

次に、本研究で採用する無線通信規格を検討する。[7]では、5Gを利用して水中ドローンが撮影する高画質な映像をリアルタイムで伝送する実験を行っている。しかし、5Gは高コストである上に消費電力が大きく小型バッテリーを使用する本研究に適用することはできない。

[8,9]では、水槽中に設置してある水質センサのデータをWi-Fi経由で収集し、リアルタイムでモニタリングしている。通信モジュールは地上に設置しており、水中にあるセンサにはケーブルで電力を供給している。また、[10]では、海にある牡蠣の籠の動きと水質センサをWi-Fi経由で陸地に送信し、そこからLTEを経由することで、クラウド上で牡蠣の成長に最適な環境を分析している。しかし、Wi-Fiも消費電力が大きいため適用することはできない。

[11]では、ZigBeeの低消費電力を活用した太陽光発電と商用電力を組み合わせたスマート灌漑システムを開発することによって、リアルタイムな土壌水分センサデータに基づいた自動制御を行い、生産性を向上させた。

ZigBeeは低消費電力であるため、本研究での利用に適しているが、水中で使用した例が少ない。IEEE 802.11を使用して水中での無線通信の性能評価を行った論文[12]では、淡水中に2台のアンテナを入れて水中通信の状態を監視することによって、特に温度と通信距離可能な関係性を明らかにしている。電磁波の伝播特性が変化することによって、16℃では22.5cm、22℃では26cmという結果が示されている。ZigBeeはIEEE 802.11と同じ2.4GHz帯で動作するため減衰の程度は同等と考えられるが、より送信電力が小さいため実際の通信可能距離は短い。しかし、水槽から外までなら届く可能性がある。そのため、予備実験を行った上で牡蠣養殖支援への適用を考える。

以降、2章では牡蠣養殖の過程について、3章では水中からの通信可能範囲を定量的に測定した予備実験とその結果について述べる。4章では予備実験の結果を踏まえて、無線マルチホップネットワークの提案と実装を行い、その実験結果について述べる。5章では本研究のまとめについて述べる。

第2章 牡蠣養殖の過程

牡蠣を養殖する過程 [13] について説明する。施設内の母貝を水槽で 12 月上旬から 1 月下旬までの約 2 カ月間、20～22℃で飼育期間中の日に平均水温を累計した積算水温が 600℃になるまで飼育し、成熟させた牡蠣を母貝として用いる。

人工受精は 2 月上旬に行い、水温 25～28℃で 14～16 日間、幼生の飼育を行う。受精後、一定の期間は浮遊幼生として水中を漂いながら、さまざまな成長段階を経て、粉碎したカキ殻に稚貝を付着して、稚貝となる。この後、稚貝が殻高 3～5 ミリメートル以上の出荷サイズになるまで施設内で育成を行う。これを中間育成という。採苗は採苗用水槽内にふるいを設置し、その上に粉碎したカキ殻を敷き、付着直前の幼生を収容して行う。沈着は 1 週間程度で終了し、その後中間育成に移っていく。

幼生時には植物プランクトンを与える。餌料である植物プランクトンの培養は、通常の栄養塩と通気、照明に加え、プランクトンの光合成を促進させるため、二酸化炭素を添加して行う。

中間育成は中央の角形水槽を挟み、片面 13 基、計 26 基の 6 トンの円形の循環水槽からなり、この円形水槽で水温は、25℃で約 40 日間育成する。水槽の底部に給水し、上部から排水する。餌料は循環水に自動的に添加され給餌される。中間育成が終了して 5 月中旬から 3～5 ミリメートル程に成長した稚貝は、養殖用種苗として地元の漁協に販売される。その後、各漁家で 1 ミリメートル目合いの養殖用籠に収容し、海での養殖を開始する。

この育苗の過程にセンサと通信モジュールを導入することで、養殖環境の状態をリアルタイムで把握できるだけでなく、水質が基準値を超えた場合には、ポンプやフィルターが自動的に制御されて水の交換や酸素の供給が行われる自動制御システムを検討する。これにより、人手による介入を最小限に抑えながら、適切な養殖環境を維持することができる。

第3章 予備実験

3.1 センサと通信モジュールの一体化

本研究では、水質センサと通信モジュールを一体化した完全防水のデバイスを水中に沈める方法を提案する。構成要素としては防水センサ、通信モジュール、バッテリー、防水ケースから成り立っており、有線接続が不要で、水中の適当な場所に設置するため大幅な設置作業の簡素化というメリットがある。一方で、水中での無線通信は信号減衰や障害物の影響を受けやすいというデメリットがある。そのため、水中からだと通信できない可能性がある。

水中からの無線通信を可能にする2点の方法について検討する。1つ目は、一体化したセンサと通信モジュールの付近に更に複数個の通信モジュールを Router として沈める方法である。水中で電波強度が小さくなったとしても、水中で無線マルチホップネットワークを構築して、データを転送することができれば、データを水槽の外に届けることができる可能性がある。しかし、Router の台数によってコストが大きくなる。2つ目は、導波管の代わりになるような円筒形容器を用意して、その中に一体化したセンサと通信モジュールを設置する方法である。そして、吸盤を用いて円筒形容器を水槽の内壁に固定して通信を行う。これにより、水の影響を最小限に抑えつつ、水中でセンシングしたデータを伝送できるのではないかと考える。

以上、提案したセンサと通信モジュールを一体化した方法は、従来の有線接続と比較して設置作業が大幅に簡略化され、設置場所の自由度が広がると期待する。ZigBee を水中で使用した例が少ないため、水中からの通信可能範囲を測定する予備実験を行うことで、センサと通信モジュールを一体化する方法と円筒形容器の導入の実用性を示して、育苗を支援するシステムを検討する。

3.2 実験端末

水中からの通信可能範囲を定量的に測定し、育苗を支援する無線マルチホップネットワークを検討するための材料を得るために予備実験を行った。そのために、ダミーデータを ZigBee で送受信する無線ネットワークを構築した。図 3.1 のようにノードを2つ用意して、1つを End device、もう1つを Coordinator として動作させるようにした。End device は一体化した

バッテリー、通信モジュールで構成されており、Coordinatorは通信モジュールとPCで構成されているようにする。通信モジュールはRaspberry Piと無線通信機器で構成する。バッテリーはRaspberry Piのモデルに応じて選択する。Raspberry Pi 3 Model B+にはPiSugar 3 PLUS[14]を、Raspberry Pi 5には市販のモバイルバッテリーを使用した。またRaspberry Piに接続されてある無線通信機器として入手簡易なMONOSTICK (JN5169) [16]を採用した。センサからRaspberry Piは、1つのバスで複数のデバイスと接続でき多種のセンサを簡単に追加可能なI2C通信でデータを送信する。Raspberry PiからJN5169は、1対1対応でノイズに強く信号がより安定しやすいUART通信で送信する。そして、ノード間はZigBee及びIEEE802.15.4でデータを送信して、Coordinatorで受信後、UART通信で送信されて、モニタで確認できる。

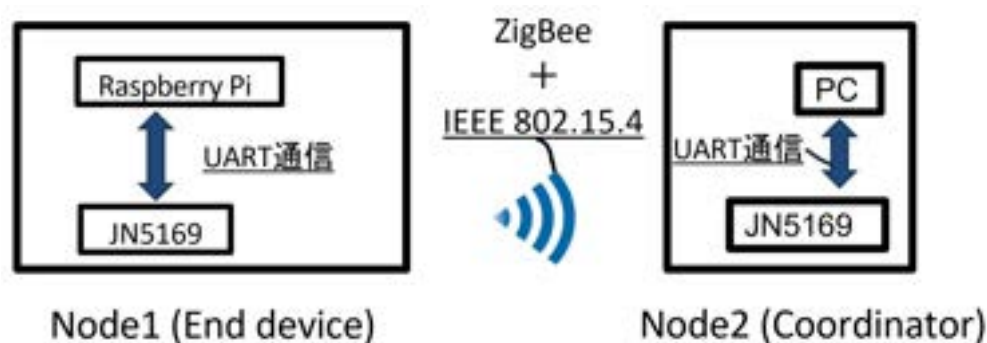


図 3.1: 無線センサネットワーク全体のデータの流れ

予備実験では、センサを使用せずにRaspberry Piで実際のセンサデータに似たダミーデータを生成して実験を行った。そのため、Raspberry Piを起動した後、自動でプログラムを実行するように構築する必要がある。

Raspberry Piでダミーデータを生成し送信するには、温度センサデータを格納していた変数dataに直接ダミーデータを記述する。プログラムではシリアル通信を使用するためのコードを記述する必要がある。1秒間に送信されるビット数をser.baudrate = 115200とし、各データフレームが8ビットのデータを含むことを表すser.bytesize = 8、データの誤り検出しない設定を表すser.parity = 'N'、各データフレームの終わりを示すser.stopbits = 1、シリアルポートでデータを受信するまで永遠に待機させるser.timeout = Noneを記述しておく。

その後、data = 0x01a6のように16進数で代入する。16進数のデータはCoordinatorで受信して可視化するとき温度に変換して表示させる。センサデータは2バイトであるのに対して、送信先であるJN5189で使用するUART通信では1byteずつしか受信することができない。そのため、シリアル通信で送信する際に2つのバイトに変数dataを分ける必要がある。上位バイトを格納する変数data0にはdata0 = (data >> 8) & 0xFFとして格納する。そして下位バイトを格納するために変数data1にはdata1 = data & 0xFFとして格納する。最後にser.write関数でser.write(bytes([data0])), ser.write(bytes([data1]))として温度データを送信する。その時、引数は整数を1バイトのバイナリデータに変換するようにbytes([data0])

とする必要がある。

自動でプログラムを実行するには、systemd というサービス管理デーモンを使用した。Raspberry Pi で、サービスファイルを作成するために `sudo nano /etc/systemd/system/your_service_name.service` を実行し、作成したサービスファイルに自動実行するための内容を記述した。その際、Raspberry Pi のシステム名や実行したいファイルのパスに注意する必要がある。新しいサービスを認識させるためのコマンドとして、`sudo systemctl daemon-reload` を実行し、サービスの有効化のために `sudo systemctl enable your_service_name`、起動するために `sudo systemctl start your_service_name` を実行した。そして、サービスが正しく起動しているかの確認するには、`sudo systemctl status your_service_name` を実行し、active running と出力されると、Raspberry Pi を起動させると自動でプログラムを実行するようになる。サービスの無効化と停止させるには `sudo systemctl disable your_service_name` と `sudo systemctl stop your_service_name` を実行する。

Raspberry Pi から送信されたセンサデータを End device の JN5169 で受信するために、u8AHLUartReadData 関数を使用して、UART 通信で 1byte ずつ受信した。事前に指定していたセンサデータの先頭を表すコマンドを認識した後に、処理を進めるように実装した。本研究では、AA がセンサデータの先頭である。その後は、図 3.2 に示すように、受信したデータがセンサの種類を表す 1byte の Sensor Number であれば、次にセンサデータの通し番号を表す 1byte の Packet Handle を受信し、その後、2bytes の Sensor Data を受信するようなアルゴリズム設計にした。これらの 4bytes の受信データを 1 つの配列に格納する。また、センサデータの先頭を表すコマンドは破棄するようにした。

| Sensor Number | Packet Handle | Sensor Data |
|---------------|---------------|-------------|
| (1byte) | (1byte) | (2bytes) |

図 3.2: 無線通信で使用するデータ構造

受信したデータを格納している配列は、PDUM_u16APduInstanceWriteNBO 関数を使用し、第 1 引数の hAPduInst に格納する。ここで、第 3 引数は格納するデータの型を指定する必要がある。本研究は 16 進数のセンサデータを扱っているため、a\xnn を使用し、nn は配列の大きさを表すため、a\x10 とした。その後、ZPS_eAplAfUnicastIeeeDataReq 関数を使用して、Coordinator の JN5169 に送信する。第 1 引数に hAPduInst になり、第 5 引数は送信先である Coordinator の JN5169 の MAC アドレスを指定する。この時、MAC アドレスは 64bits である。eSecurityMode は、セキュリティ無効を表す ZPS_E_APL_AF_UNSECURE にしている。例えば、ブロードキャスト通信をする場合は、ZPS_eAplAfBroadcastDataReq 関数を使用し、第 5 引数は全端末に送信することを表す ZPS_E_BROADCAST_ALL を指定すればよい。

送信するタイミングは、End device 特有のスリープ機能を活用してセンサデータを送信している。本研究では、5 秒に 1 回スリープ状態から覚めるため、そのタイミングでデータを Coordinator へ送信しているようにした。スリープ機能の時間指定はアプリケーションによって変更する必要がある。

Coordinator の JN5169 は Enddevice からデータを受信すると、Coordinator の app_endpoint.c の APP_vtaskMyEndPoint 関数にある sStackEvent.eType の ZPS_EVENT_APS_DATA_INDICATION で処理をする。INDICATION とは、他端末からデータを受信したことを表す status である。そして、PDUM_u16APduInstanceReadNBO 関数を使用し、sStackEvent.uEvent.sApsDataIndEvent にある受信データを読み込む。sStackEvent.uEvent.sApsDataIndEvent.hAPduInst はデータそのものを表しており、sStackEvent.uEvent.sApsDataIndEvent.uSrcAddress.u16Addr は送信元の short address を表している。そのため、PDUM_u16APduInstanceReadNBO 関数の第 1 引数に sStackEvent.uEvent.sApsDataIndEvent.hAPduInst、第 2 引数ではどの位置から読み取るのかを表す位置、第 3 引数には受信データを格納する配列する。

そして、受信データを PC 上に表示するために、UART 通信で表示したいデータを送信する。vAHL_UartWriteData 関数を使用し、第 1 引数には DBG_E_UART_0 を、第 2 引数には送信したいデータを指定する。例えば、送信元である End device の short address を送信したい場合は、vAHL_UartWriteData(DBG_E_UART_0, sStackEvent.uEvent.sApsDataIndEvent.uSrcAddress.u16Addr) とすれば良い。その後、memset 関数で受信したデータを格納する配列を初期化する。その後、データを PC で受信して処理することで、センサデータを確認できる。

具体的なデータ構造としては、どのノードから受信したのかを表す Short Address、電波通信品質を示す Link Quality Indicator (LQI) [16] をパケットに付与して、USB 接続で接続されている PC に UART 通信で送信した。LQI とは電波通信品質を示す値であり、0 から 255 までの数値で表される。評価区分として、50 未満が悪い、50～100 がやや悪い、100～150 が良好、150 以上はアンテナの近傍である。図 3.3 の破線で囲んでいる部分が受信データに追加した箇所である。

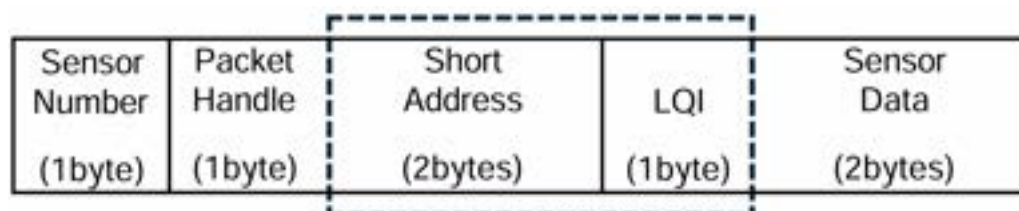


図 3.3: UART 通信で使用するデータ構造

3.3 水中での通信品質評価

通信可能範囲に水がどれほど影響しているのかを定量的に測定するために、水以外の障害物がない環境を用意した。また、コストや設置の容易さを重視したため、軽量かつ電波伝播に与える影響が少ないという利点をもつ防水ケースとして簡易的なプラスチック製のジッパー袋を採用した。

水中距離を変化させるために図 3.4 と図 3.5 のような高さ 0.45m、幅 0.60m の水槽を用いて電波強度の測定を行った。実験内容として、図 3.6 に示すように Coordinator は水槽に沈めずに、End device だけを水槽に沈めており、Coordinator までの距離は 2.0m に固定にし、図 3.7 にある x を 0.05m、0.10m、0.15m、0.20m と変化させた。

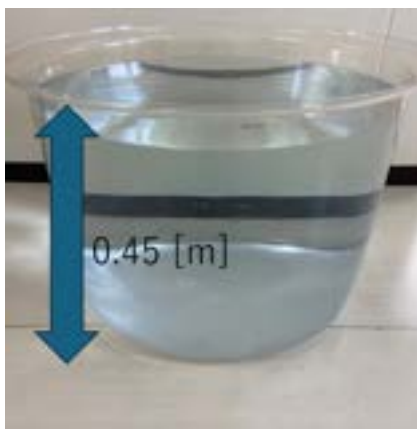


図 3.4: 水槽の高さ

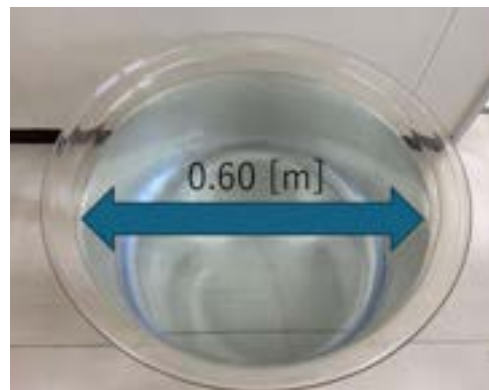


図 3.5: 水槽の幅

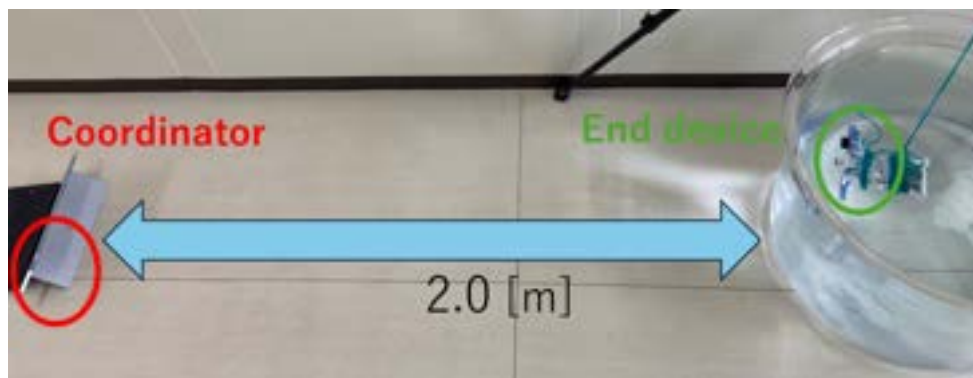


図 3.6: 水槽外の通信距離

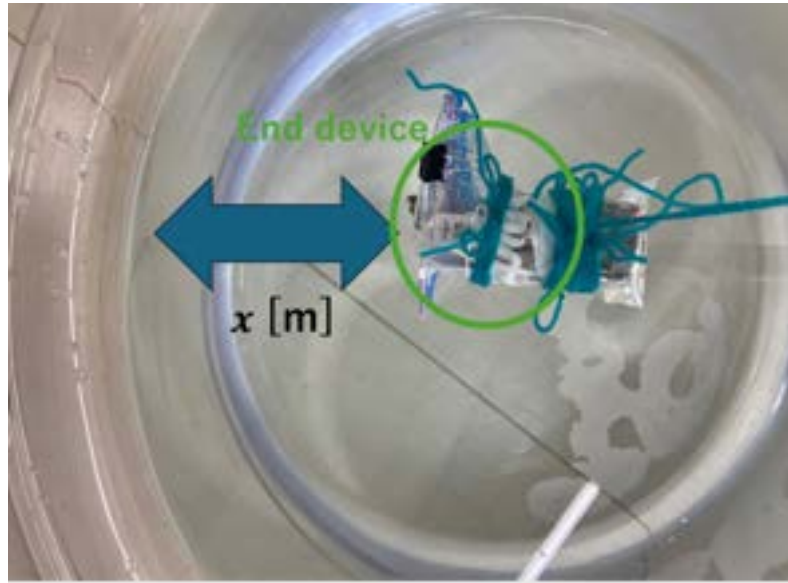


図 3.7: 水中部分の通信距離

実験結果を図 3.8 に示す．縦軸は LQI 値と変換式 (3.1) で受信信号強度に変換した値 P を示し，横軸はメートル単位の水中距離を表している．また，○印は 10 パケットの平均値を表しており，×印は 10 個の○印の平均値を表している．

$$P = \frac{7 \times \text{LQI} - 1970}{20} \quad [\text{dBm}] \quad (3.1)$$

0.05m の位置では，LQI の評価区分「悪い」という電波通信品質になっており， x の距離が大きくなるにつれて，LQI が悪くなっていることが明らかになった．そして，0.20m の位置では，データを受信不可能となった．

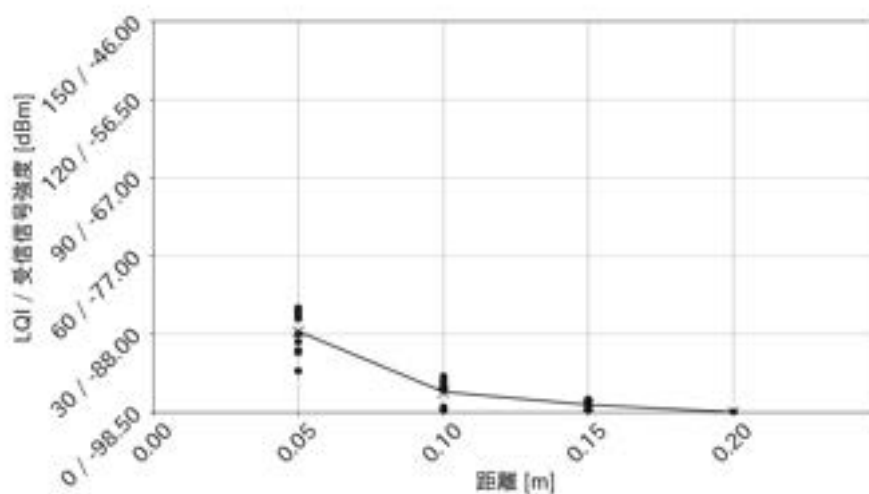


図 3.8: 水中距離を変化させた場合の LQI 及び受信信号強度

また、パケットロス率を図 3.9 に示す。縦軸はパケットロス率を表しており、横軸はメートル単位の水中距離を表している。0.05m の位置ではパケットロス率はほぼ 0 であるが、 x の距離が大きくなるにつれて、パケットロス率が大きくなっていることが明らかになった。0.15m では約 90% のパケットロス率が発生し、0.20m では 100% のパケットロス率となった。

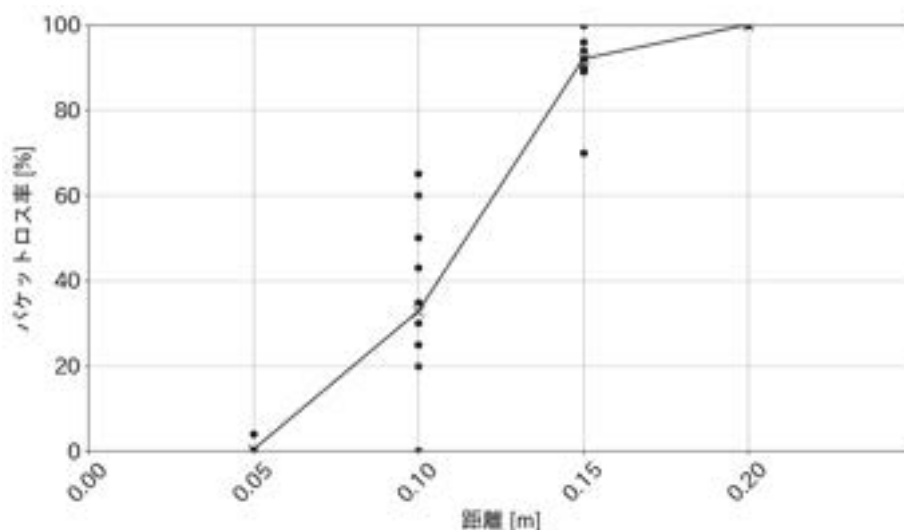


図 3.9: 水中距離を変化させた場合のパケットロス率

ここで図 3.10 で示すような円筒形容器を導入する。円筒形容器には水が入らないようにする必要があり、底には吸盤を設置して、図 3.11 のように水槽の内壁に設置する。設置場所としては、パケットロス率が 100% で受信不可能であった 0.20m を水面からあるようにした。そして、図 3.12 に示すように、円筒形容器内にある End device も内壁から 0.20m になるよう設置する。



図 3.10: 円筒形容器の全体図



図 3.11: 円筒形容器を水中に沈めた状態 (横から)

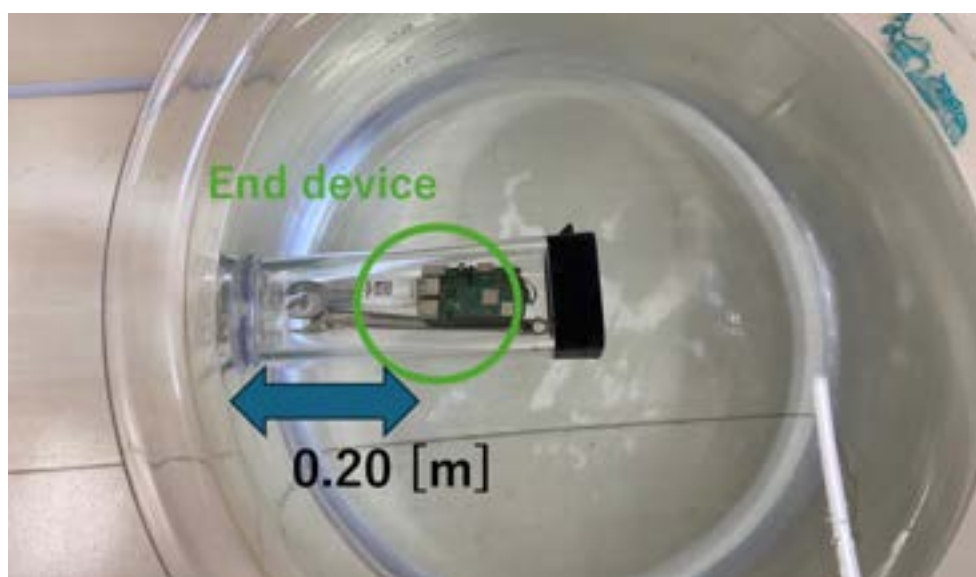


図 3.12: 円筒形容器を水中に沈めた状態 (上から)

図 3.13 に測定結果を示す。縦軸は LQI 値と受信信号強度 P ，横軸が時間経過を表している。円筒形容器を導入していない場合は 0.20m の地点で受信不可能であったが，円筒形容器を導入することで，平均 LQI 値は 68.38，平均受信信号強度は -74.55 となり，パケットロスなく受信することができた。したがって，円筒形容器の実用性を示すことができた。また水中からの通信のため，電波は内壁にある吸盤の直線方向しかないので，水中に沈めない Line of Sight と比べると，LQI 値は低くなることに注意する。

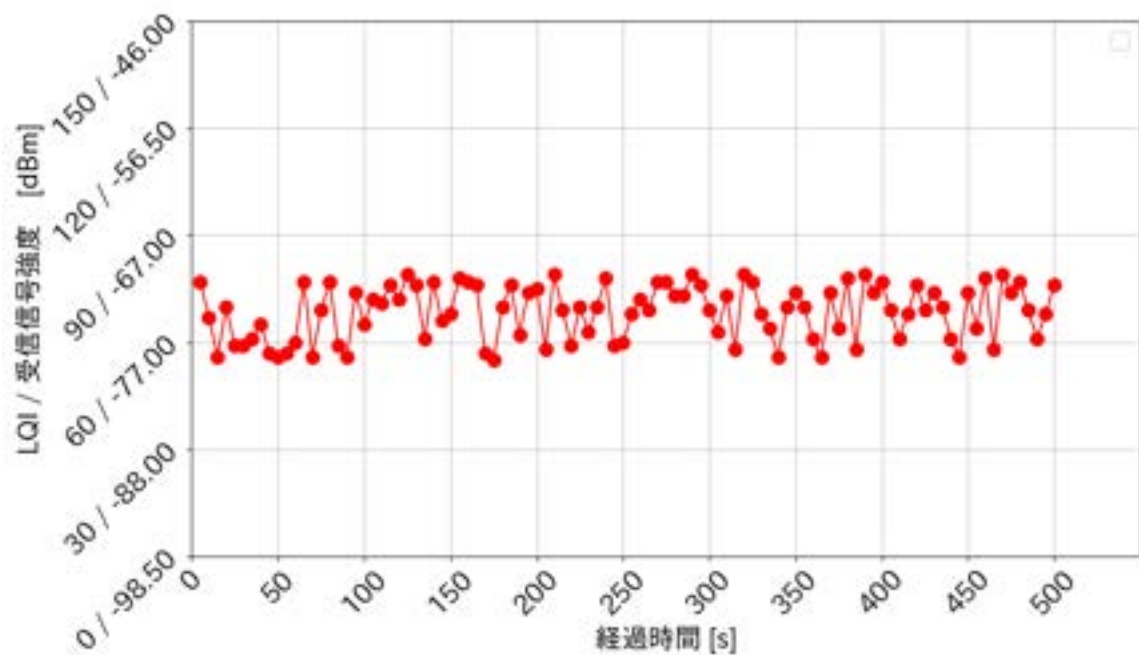


図 3.13: 円筒形容器を用いた場合の測定結果

また、水分子による吸収や散乱の影響で電波強度が低下するため、一定距離を超えた Coordinator へのデータ送信には限界がある。そこで、水中からの通信可能範囲を測定する実験結果より、円筒形容器の無い状態で実用的に許容されるパケットロス率であった 0.1m 以内の距離に End device を固定して、Coordinator への通信可能範囲を測定した実験を行った。始めに、水による影響と比較しやすいように、始めに見通しの良い場所で、Coordinator と End device との距離を変化させて通信評価を行った。図 3.14 の x が変化させた距離である。また、JN5169 の電波の指向性より、End device は Coordinator のある方向を向いているように配置した。

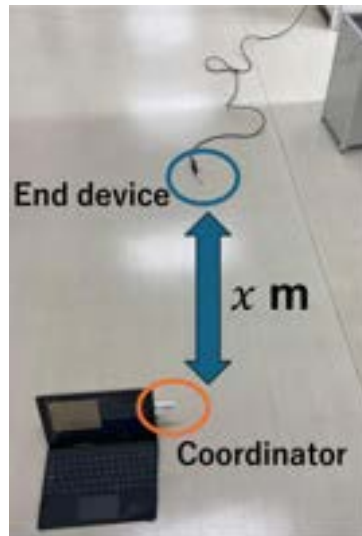


図 3.14: 障害物のない見通しの良い場合

次に, End device としての1つのノードを図3.15, 3.16に示すようにいずれの方向にも 0.05m 以上 0.10m 以内の水中伝播距離があるように水槽に沈め, 水に入っていない Coordinator へデータを送信させた.

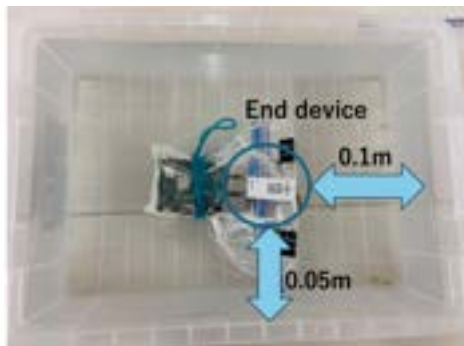


図 3.15: End device を水に沈めた場合 (上から)

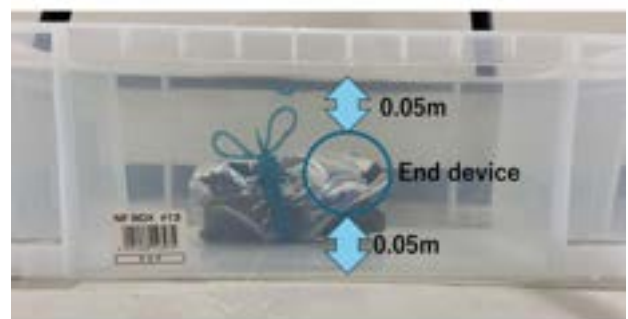


図 3.16: End device を水に沈めた場合 (横から)

図 3.17 は Coordinator も含めた全体的な様子である. 水槽から Coordinator までの距離 x を変化させて実験を行った.

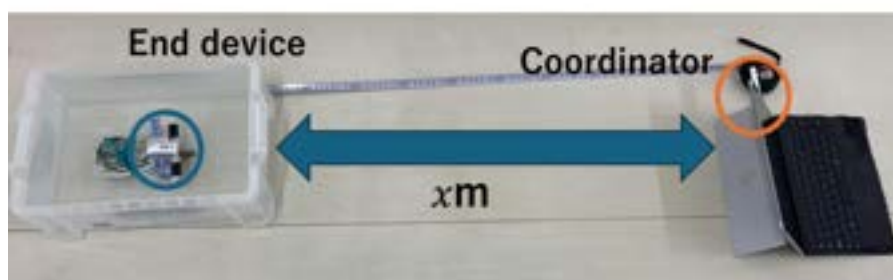


図 3.17: 全体図

最後に，ノードを2つとも水中に投下した場合の通信品質評価を行った．図 3.18 に示すように，水槽間の距離 x を変化させ，End device から Coordinator へとデータを送信した．

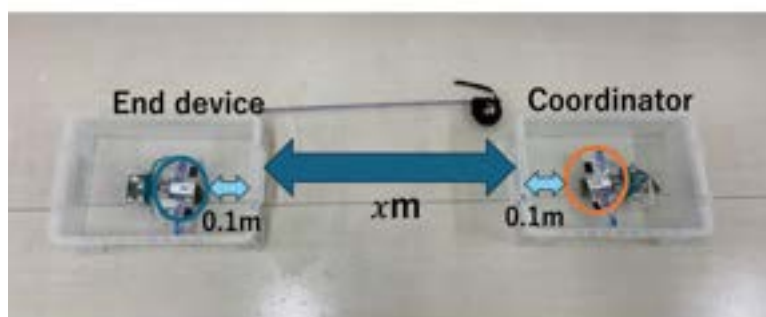


図 3.18: 2つのノードを水に沈めた場合

図 3.19 が通信評価結果である．赤色のグラフが“見通しの良い場所での通信評価”，青色のグラフが“1つのノードだけを水中に投下した場合の通信評価”，紫色のグラフが“2つのノードとも水中に投下した場合の通信品質評価”，を示している．10 パケットを送受信しその平均値が1つのプロットであり，更にそれを各距離 10 回測定した平均値を結んでいる．青色で示した“1つのノードだけを水中に投下した場合の通信評価”では，6m の距離で Coordinator にデータが届かず，LQI 値を 0 としてそれ以降の距離は測定していない．紫色で示している“2つのノードとも水中に投下した場合の通信品質評価”の 1m 以降も同様である．グラフより，水中から 5.5m までは通信可能だということが確認された．さらに水中に投下したノード間では，0.75m まで通信可能範囲ということも確認できた．

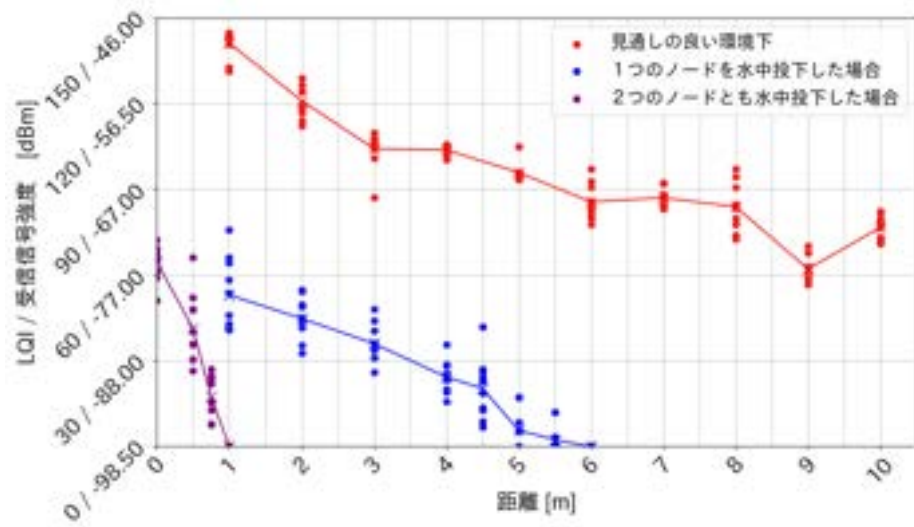


図 3.19: 2つのノード間での LQO 及び受信信号強度

次に、図 3.20 に各実験でのパケットロス率を示す。

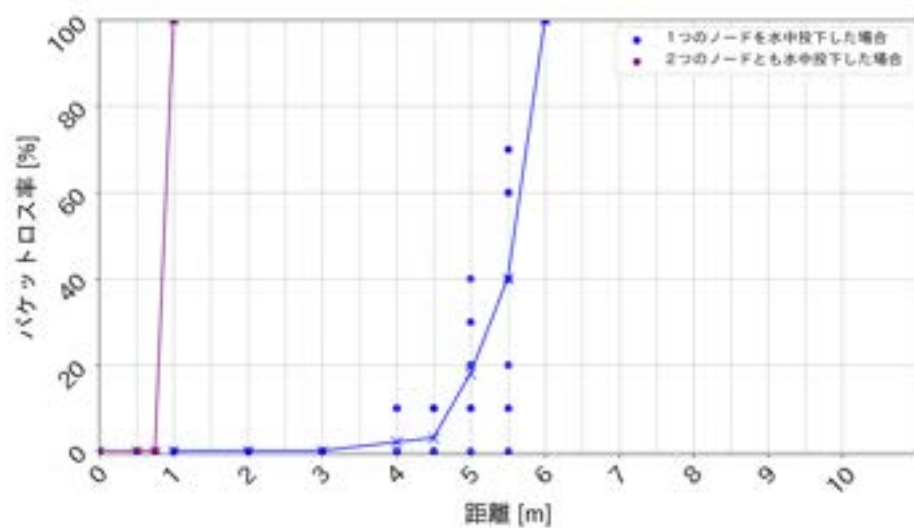


図 3.20: 2つのノード間でのパケットロス率

“1つのノードを水中に投下した場合”の実験では、5.5mまで通信可能距離であったが、約20%のパケットロス率が発生しており、6m地点ではパケットロス率が100%になった。また、“見通しの良い場所での通信評価”である実験では、パケットロスが発生しなかったため、グラフに示していない。

3.4 養殖場でのマルチホップ通信確認

本研究で構築した無線マルチホップネットワークにより、数多くの水槽がある環境下でも、センサデータを受信することが可能かどうかを確認するために、養殖場で実験を行った。養殖場の環境としては、奥行き 10m、幅 5m、高さ 2.5m ほどの部屋が 2 つあり、予備実験で使用した水槽の約 1.5 倍の水槽が、部屋の両サイドに 2 段になって約 13 台、中央に 6 台、計約 32 台の水槽が設置してある。そこで、部屋の奥にある水槽の裏に End device1 台を設置して、End device から部屋の出入り口までに適当な感覚で Router3 台、もう片方の部屋に Coordinator1 台を設置して無線マルチホップネットワークを構築した。また、実際に温度センサを使用した。結果、図 3.21 で示すように Router を経由してマルチホップ型で温度データを伝送して、Coordinator でパケットロスなく受信することができた。温度データには通し番号を付与しており、その番号でパケットロスの有無を確認した。



```
Serial: (COM5, 115200, 8, 1, None, None - CLOSED) - Encoding: (ISO-8859-1)
APP: Network Started
APP: Channel - 11
APP: vCheckStackEvent: unhandled event 29
APP: vCheckStackEvent: unhandled event 29
01. [MAC: 0xc29d][Data16: 0x007b14ae47e17a00]
02. [MAC: 0xc29d][Data16: 0x00e3a59bc420b000]
03. [MAC: 0xc29d][Data16: 0x007f6abc74931800]
04. [MAC: 0xc29d][Data16: 0x00d578e926310800]
05. [MAC: 0xc29d][Data16: 0x007b14ae47e17a00]
06. [MAC: 0xc29d][Data16: 0x00df4f8d976e1200]
07. [MAC: 0xc29d][Data16: 0x002fdd2406819500]
08. [MAC: 0xc29d][Data16: 0x0025068195438b00]
09. [MAC: 0xc29d][Data16: 0x00cff753e3a59b00]
0a. [MAC: 0xc29d][Data16: 0x00c74b3789416000]
0b. [MAC: 0xc29d][Data16: 0x002fdd2406819500]
0c. [MAC: 0xc29d][Data16: 0x0079e9263108ac00]
0d. [MAC: 0xc29d][Data16: 0x0025068195438b00]
0e. [MAC: 0xc29d][Data16: 0x00dd240681954300]
0f. [MAC: 0xc29d][Data16: 0x002db29defa7c600]
10. [MAC: 0xc29d][Data16: 0x0079e9263108ac00]
11. [MAC: 0xc29d][Data16: 0x00d9cef753e3a500]
12. [MAC: 0xc29d][Data16: 0x00cff753e3a59b00]
13. [MAC: 0xc29d][Data16: 0x0079e9263108ac00]
14. [MAC: 0xc29d][Data16: 0x008d976e1283c000]
15. [MAC: 0xc29d][Data16: 0x007b14ae47e17a00]
16. [MAC: 0xc29d][Data16: 0x00295c8fc2f52800]
17. [MAC: 0xc29d][Data16: 0x007d3f355eba4900]
18. [MAC: 0xc29d][Data16: 0x0083c0caa145b600]
```

図 3.21: Coordinator で受信した温度センサデータの通し番号

第4章 本実験

4.1 無線マルチホップネットワークの提案

水槽との距離を変化させた場合の電波強度の測定結果を踏まえて、Router を End device と Coordinator の間に設置することで、水中からの End device のデータを水槽外の Coordinator に送信できるようにする無線マルチホップネットワークを提案する。無線マルチホップネットワークを構築することで、End device と Coordinator との距離が離れていたとしても、Router を経由することで、水中にある End device からのデータを Coordinator で受信できるのではないかと考える。図 4.1 と関連付けると Node1 が End device, Node2 が Router, Node3 が Coordinator である。

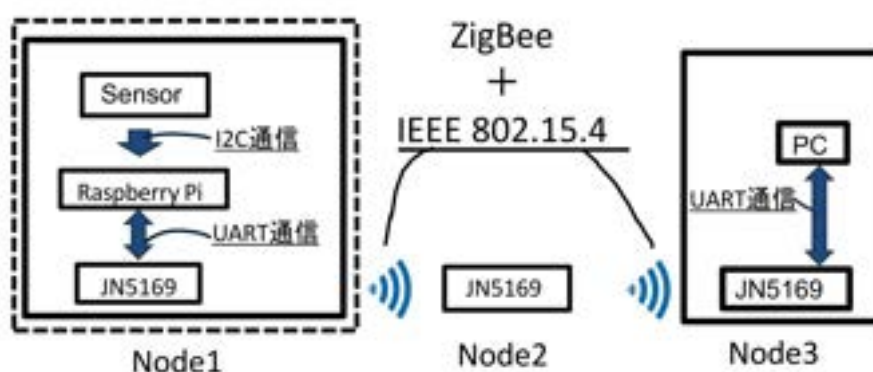


図 4.1: ネットワーク全体のデータの流れ

4.2 無線マルチホップネットワークの実装

提案したシステムを温度センサを扱えるように実装した。温度センサは図 4.2 に示すように、防水機能のある温度センサ DS18B20[15] である。また、プルアップ・プルダウン抵抗のアダプタを使用するため、ブレッドボードとジャンパワイヤは使用しない。DS18B20 には 3 本のピンがあり、3.3V または 5.0V の電源を供給するピン VDD、グランド接続用ピン GND、データ通信用ピン DQ である。本研究では、VDD を Raspberry Pi の 3.3V ピン、GND を GND ピン、DQ を GPIO ピン 4 に接続した。DQ は他のピンでも可能である。Raspberry Pi で DS18B20 を認識させるための設定を行う。1-Wire インターフェースを有効化するため

に、`$ sudo raspi-config` を実行して設定ツールを開く。Interface Options から 1-Wire を選択して有効化する。その後保存して、Raspberry Pi を再起動する。

次に 1-Wire 用のカーネルモジュールを手動で有効化する。再起動後、`$ sudo modprobe w1-gpio` と `$ sudo modprobe w1-therm` を実行することでモジュールをロードする。DQ を GPIO ピン 4 に接続しているため、`$ sudo nano /boot/config.txt` 実行して、`dtoverlay=w1-gpio-pullup,gpiopin=4,pullup=4` を config.txt の末尾に追加する必要がある。最後に DS18B20 が認識されているか確認する。`$ ls /sys/bus/w1/devices/` を実行して、出力結果に 28-で始まるディレクトリが表示されれば DS18B20 が認識されていることになる。

Raspberry Pi でプログラムを組む前に、`$ pip install w1thermsensor` を実行して w1thermsensor モジュールをインストールする。開発環境によってインストールコマンドが異なるため注意する必要がある。

DS18B20 からデータを受信して、Raspberry Pi に接続してある JN5169 にシリアル通信でセンサデータを送信するプログラムの流れについて説明する。始めに、必要なライブラリを `from w1thermsensor import W1ThermSensor` でインポートする。そして、`ds18b20_sensor = W1ThermSensor()` で DS18B20 のインスタンスを作成し、温度を取得するための準備をする。

DS18B20 から現在の温度を取得するために、`temperature = ds18b20_sensor.get_temperature()` として、変数 `temperature` に代入する。そして、`print("0:.2f C".format(temperature))` で温度を小数点以下 2 桁まで表示させる。浮動小数点数はシリアル通信のようなバイト単位のデータ送信に直接適さないため、整数値に変換する必要がある。温度データを小数点以下の精度を保ちながら整数化するために、`data = int(temperature * 100)` とすることで温度データを 100 倍して変数 `data` に代入して整数化する。例えば、温度が 23.45 °C の場合、`data` には 2345 が格納されていることになる。

その後のセンサデータのシリアル通信での送信方法は 3.2 の実装方法でのダミーデータ生成と同じである。



図 4.2: 温度センサ DS18B20

4.3 無線マルチホップネットワークの実験

次に、水中に沈めた複数台の End device から伝送したデータを Coordinator で受信可能かどうか確認するために、提案した無線マルチホップネットワークを図 4.3 のような実験環境で動作確認を行った。E, R, C はそれぞれ、水中に沈めた End device, 水中に沈めていない Router, 水に沈めていない Coordinator を表している。Router と Coordinator の距離を変化させて通信評価を行った。End device と Router との距離 0.5m は本研究では固定にする。また、ここでは Raspberry Pi で生成したダミーデータを扱っている。

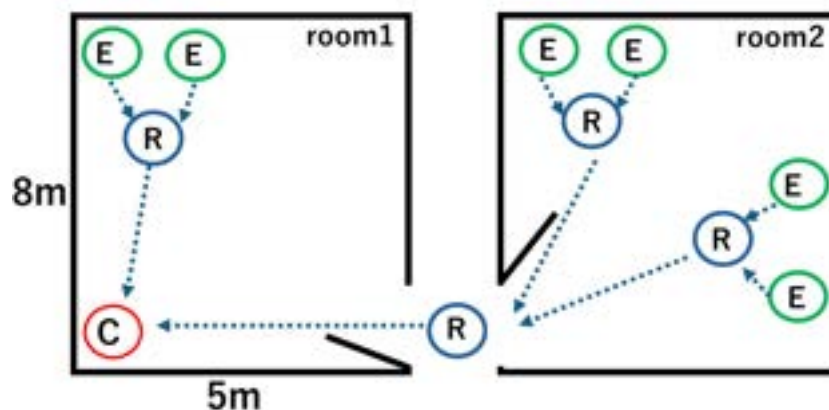


図 4.3: 構築した無線マルチホップネットワーク

実験結果として、End device と Coordinator との距離を 6m から 10m に変化させても、Router を経由することで、パケットロスなく通信可能であった。水槽との距離を変化させた場合の電波強度の測定実験では、通信可能最大距離 5.5m でパケットロス率が約 40%、6m でパケットロス率が 100%であったため、この提案システムでの実験により、無線マルチホップネットワークの有効性を示すことができた。

そして、実際に実装したノードを円筒形容器内に設置して、無線マルチホップネットワークの動作確認を行った。図 4.4 に示すように円筒形容器に穴を開け、そこから温度センサを出し、接続部分は防水になるよう加工してある。



図 4.4: 実装した DS18B20

図 4.5 にネットワーク全体を示すように、Coordinator と Router との距離は 4.0m と固定した。また図 4.6 に示すように、Router と End device との距離は 0.20m とした。

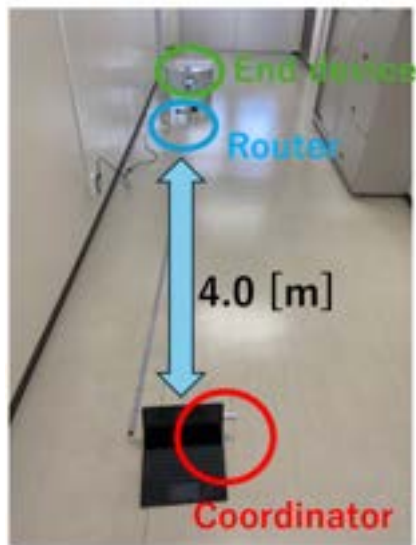


図 4.5: ネットワークの全体

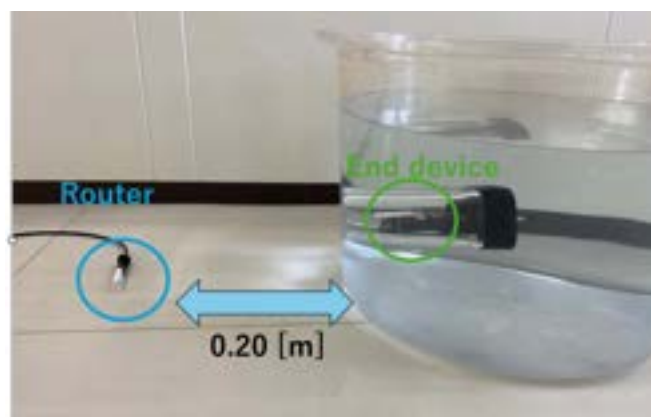


図 4.6: End device と Router の配置

結果は、パケットロスなく温度センサデータを Coordinator で収集することができた。また、平均 LQI が 106、平均受信信号強度が -61.4dBm となり、無線マルチホップネットワークを構築することで、LQI の通信評価基準としては「良い」区分になる値を受信して、無線マルチホップネットワークの実用性を示すことができた。

第5章 結論

本研究では、牡蠣の稚貝を育てる陸上の養殖施設において、無線通信を用いて水質などのセンサデータを自動的に収集するシステムについて検討した。センサと通信モジュールを一体化することで、配線のスペース確保や劣化による交換作業を削減する。そこで、ZigBeeを用いた無線マルチホップネットワークの構築を検討し、水中から通信可能範囲を測定した。結果、水中に投下して End device と Coordinator への距離を固定して、水槽の内壁と End device との距離を変化させた通信品質評価では 0.15m まで通信可能で、0.20m ではパケットロス率が 100% となった。そこで円筒形容器を導入することで 0.20m の位置でもパケットロスなく通信可能であることを示すことができた。また、円筒形容器の無い状態で実用的に許容されるパケットロス率であった 0.1m 以内の距離に End device を固定して、Coordinator との距離を変化させた通信品質評価の実験を行った。結果、End device のみ水中に沈めた場合は 5.5m まで、End device と Coordinator の両方を沈めた場合は 0.75m まで無線通信可能ということを明らかにした。そして、Router を介した無線マルチホップネットワークを構築することによって、水中に沈めていないノード間における通信の LQI 値と同等の値で、パケットロスなく送信可能にした。

現在は、1つのセンサのみ対応しているシステムであるため、今後はシステムの拡大と自由度を考慮して、複数台のセンサに対応したシステムを構築すべきである。また、本研究で使用した円筒形容器よりも小さくて細い容器を採用して養殖場で実用性を確認する必要がある。そして、牡蠣の養殖場で本研究で構成したシステムを導入にして動作確認を行うことで、システムの実用性を示す必要がある。

謝辞

本研究を行うにあたり，熱心にご指導ご教授くださいましたガジェゴス ラモネト アルベルト助教，木下和彦教授，池田健司准教授に厚く感謝します．

また，研究に対して有益な討論をしてくださいました B2 研究室の学生の皆さんに感謝致します．

参考文献

- [1] 政府統計の総合窓口, “データセット一覧,” <https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei=00500213&tstat=000001145546&cycle=7&year=20220&month=0&tclass1=000001154213&tclass2=000001212440>, 2022.
- [2] 農林水産省, “漁業生産を支える人材確保,” 2021. <https://www.jfa.maff.go.jp/j/council/seisaku/kikaku/attach/pdf/210721-2.pdf>
- [3] KDDI 株式会社, “KDDI が 取組むスマート漁業,” 2019, <https://www.kantei.go.jp/jp/singi/keizaisaisei/miraitoshikaigi/suishinkaigo2018/nourin/dai13/siryou3.pdf>
- [4] クレバリアグリ株式会社, “IoT + AI による農業の専門家のための農業クラウドサービス開始,” 2018, <https://clevagri.com/20180301press/>
- [5] 株式会社 小松製作所, “コマツのスマート林業と林業機械,” 2021, https://www.cas.go.jp/jp/seisaku/seicho/wgkaisai/nougyou_dai2/siryou2.pdf
- [6] 株式会社リブル, “スマート牡蠣養殖で未経験者でもチャレンジできる新しい形の水産業に挑戦するリブル,” 2024, https://smartiot-forum.jp/iot-val-team/iot-case/reblue?utm_source=chatgpt.com
- [7] NTT ドコモ, “5G と水中ドローンを活用した漁場遠隔監視の実証実験に成功,” Nov. 2019, https://www.docomo.ne.jp/binary/pdf/corporate/technology/rd/topics/2019/topics_191127_00.pdf.
- [8] A. Machado, J. S. Pinto, and R. Rocha, “Biodepura: bi-valve depuration methodologies adapted to the physiological needs,” 10th Euro-American Conference on Telematics and Information Systems (EATIS '20), 2021.
- [9] N. Ya'acob, N. N. S. N. Dzulkefli, A. L. Yusof, M. Kassim, N. F. Naim, and S. S. M. Aris, “Water quality monitoring system for fisheries using internet of things (iot),” IOP Conference Series: Materials Science and Engineering, vol. 1176, no. 1, p. 012016, August 2021.

- [10] 山田海輝, 辻明典, 木下和彦, “持続可能な牡蠣養殖のための環境モニタリング,” 電子情報通信学会総合大会, 2024.
- [11] A. Sagala, J. R. Sitompul, J. L. Hutauruk, and R. Sitorus, “Smart farming-drip irrigation controlled using lr-wpan with hybrid power,” Jurnal Sistem Informasi dan Ilmu Komputer Prima (JUSIKOM PRIMA), vol. 5, no. 2, pp. 135–140, 2022.
- [12] S. Sendra Compte, “Deployment of efficient wireless sensor nodes for monitoring in rural, indoor and underwater environments,” Ph.D. dissertation, Editorial Universitat Politècnica de València, 2013.
- [13] 北海道立総合研究機構, “カキ人工種苗生産（シングルシード）,” 2013, <https://www.hro.or.jp/fisheries/h3mfcd0000000gsj/marine/o7u1kr000000019q/o7u1kr0000000d760/o7u1kr0000000dcld.html>
- [14] PiSugar Kitchen, “PiSugar,” <https://www.pisugar.com/>
- [15] Analog Devices, Inc., “DS18B20,” <https://www.analog.com/jp/products/ds18b20.html>
- [16] モノワイヤレス株式会社, “MONOSTICK,” <https://www.monowireless.com/jp/products/MoNoStick/>.