

# TTNを利用したLoRaWAN無線による船舶衝突防止システムの構築

工学科電気電子コース  
通信システム工学研究室  
田中 雅玖斗

# 研究目的と内容

## 目的

- 無線中継器(ゲートウェイ)を共有する**TTN** (The Things Network)というサービスを用いて, **LoRaWAN**無線網を瀬戸内海全域に張り巡らせる
- **船舶衝突防止システム**を構築し**船舶事故を減らす**

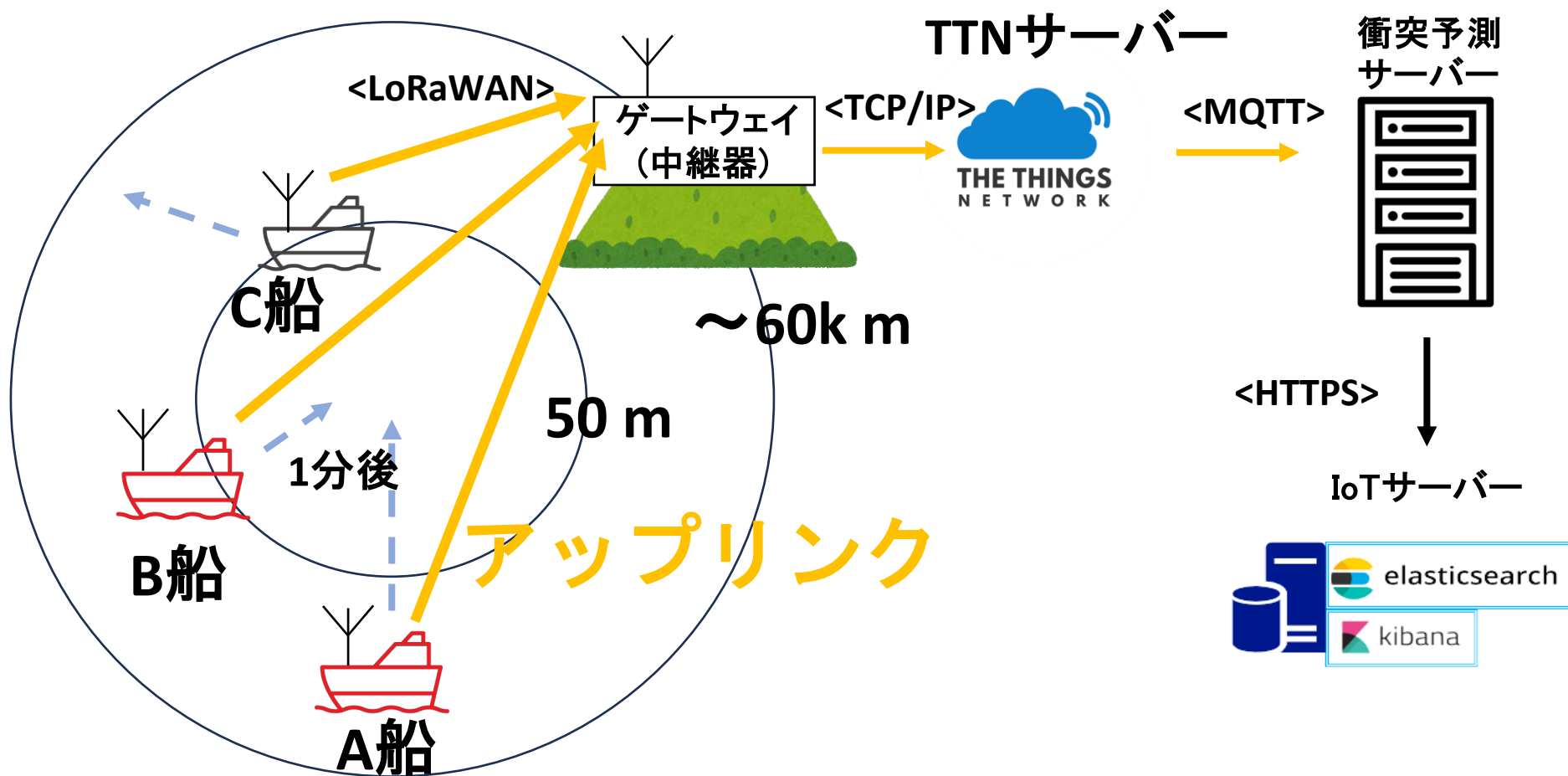
➡ 課題：TTNのダウンリンクが使えるかは不明だった

## 論文内容

1. 長距離通信に備えて, RSSI性能に基づくエンドデバイス3種の選定

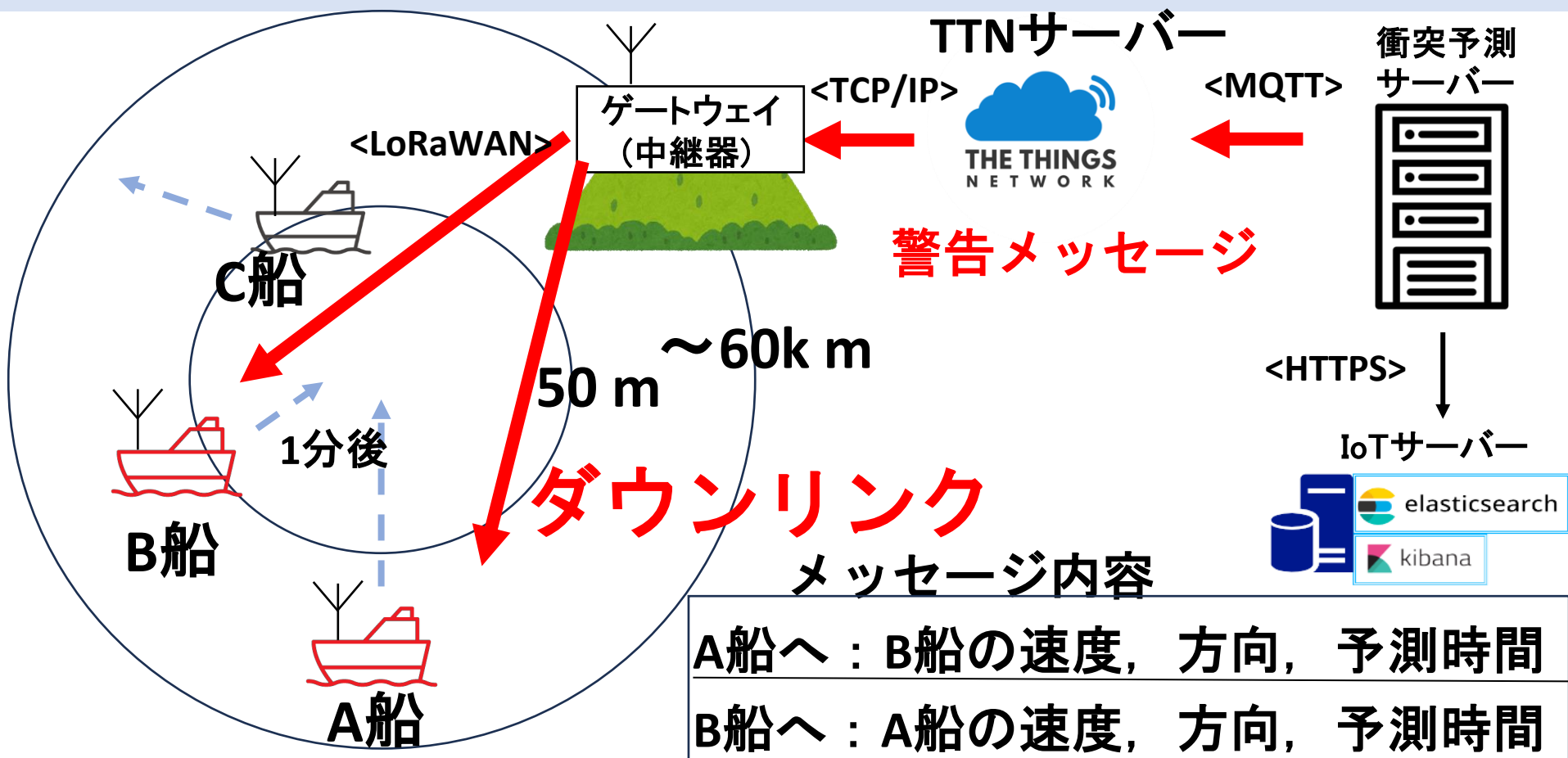
② サーバー側で衝突予測を行い, エンドデバイスにダウンリンクを使って通知するシステムを構築

# 構築した衝突防止システムの概要



**30秒毎**にアップリンクでGPS情報を送信

# 構築した衝突防止システムの概要



1分後, 50 m以内に他船がいると予測された時, 警告メッセージをダウンリンクで送信

# 1分後の衝突予測テスト

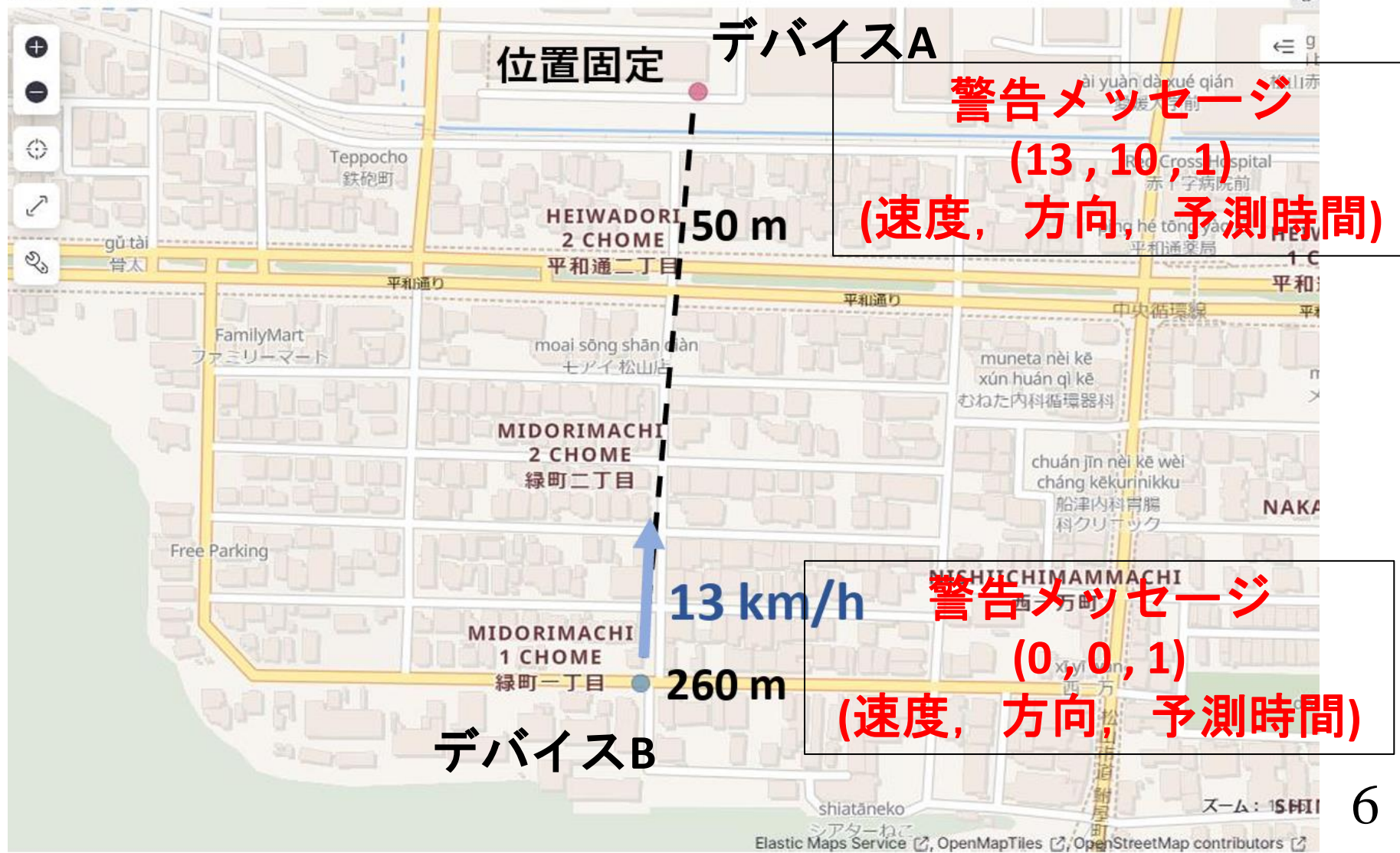
衝突予測デバイス





# 1分後の衝突予測テストの結果

衝突予測デバイス



# まとめ

## ◆結論

- ・ 衝突リスクがある船舶へ警告メッセージをTTNの**ダウンリンク**を使って通知できることを確認した

## ◆今後の課題

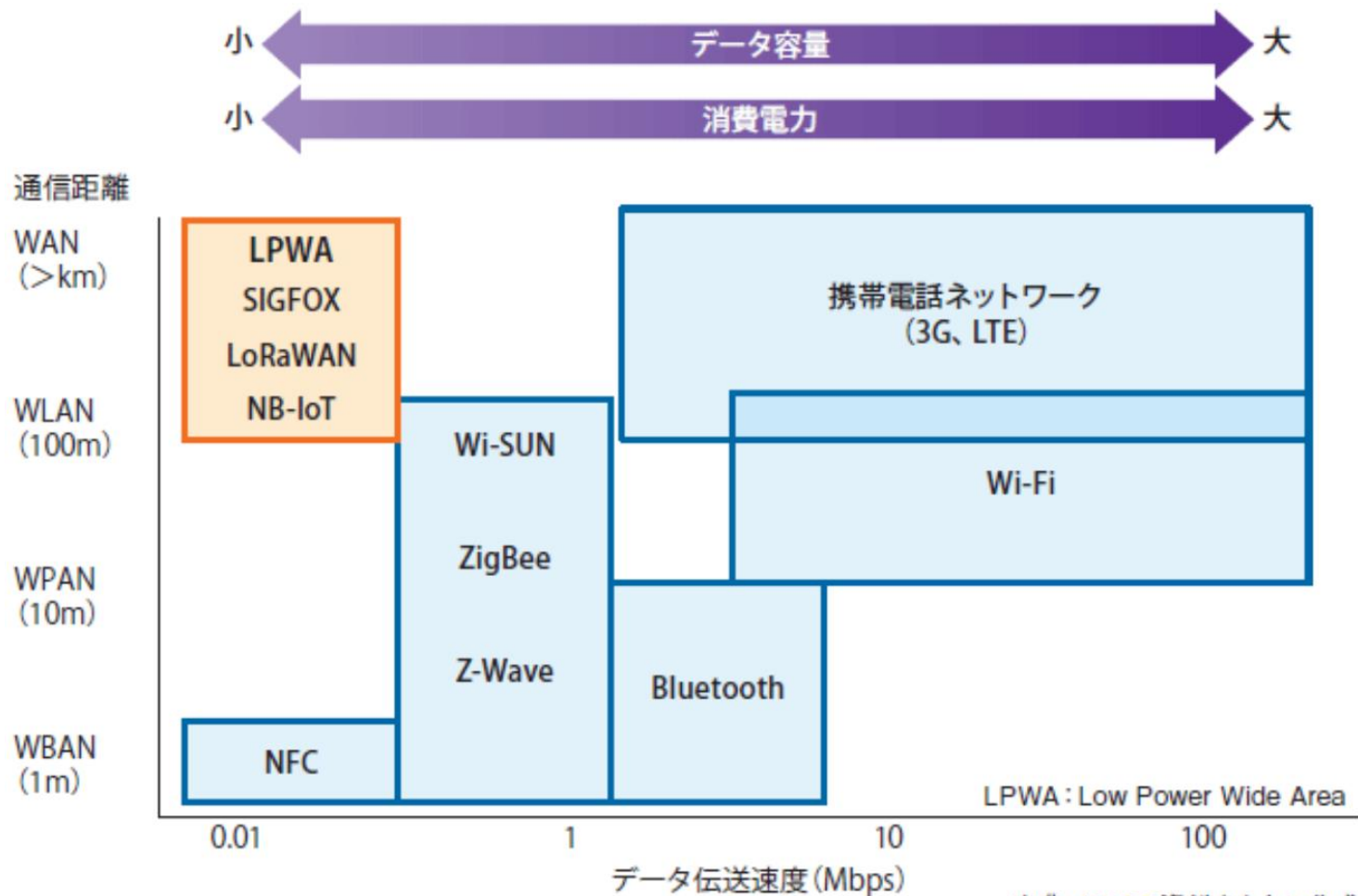
- ・ LoRaWANゲートウェイ(陸上の中継器)を設置する場所を検討し**海上運用**を行う
- ・ 衝突予測の**精度を高める**ため、エンドデバイスのGPSデータの送信間隔を可変とし、衝突リスクがあるときには**頻繁**に送信するようにソフトウェアを変更する

# 付録



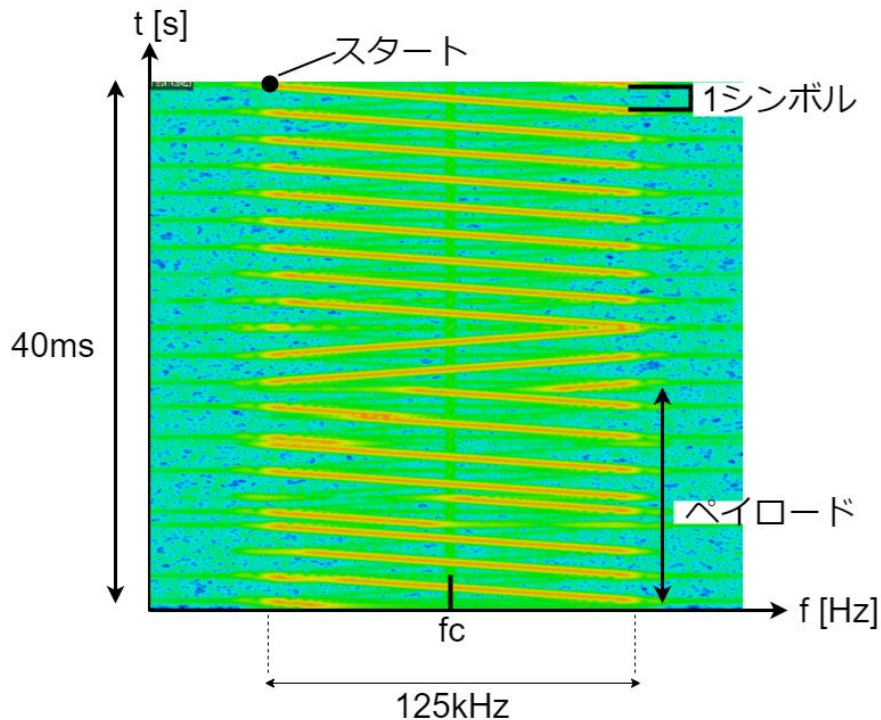
# LoRaとは？

LoRa とは？ ➡ 無線変調方式の名称



出典：KCCS資料をもとに作成

# LoRa変調



- チャープスペクトラム拡散  
周波数を連続的に変化させて拡散する  
周波数あたりの電力密度が高く、SNRを改善し遠距離通信に有利
- シンボル  
開始周波数から周波数が変化し1周して元の周波数に戻るまでが1シンボル
- ペイロード  
シンボルごとに開始周波数が $2^{SF}$ 通りの異なるアップチャープでデータをエンコード

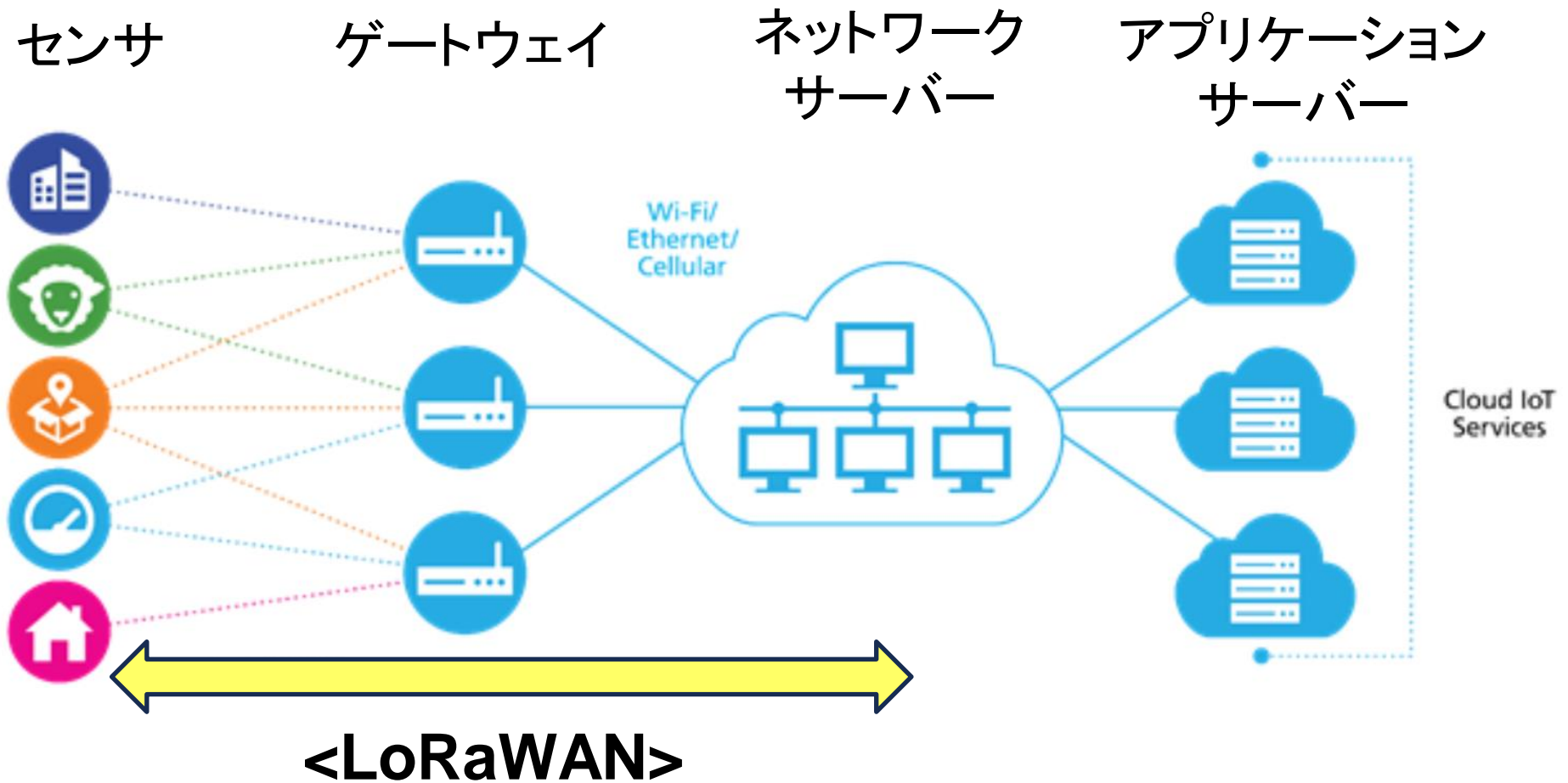
# プライベートLoRaとLoRaWANの違い

**LoRaWAN** : 非営利団体LoRaAllianceが規定した  
グローバルなプロトコル

**プライベートLoRa** : 個人又は組織が独自に設定した  
プライベートなプロトコル

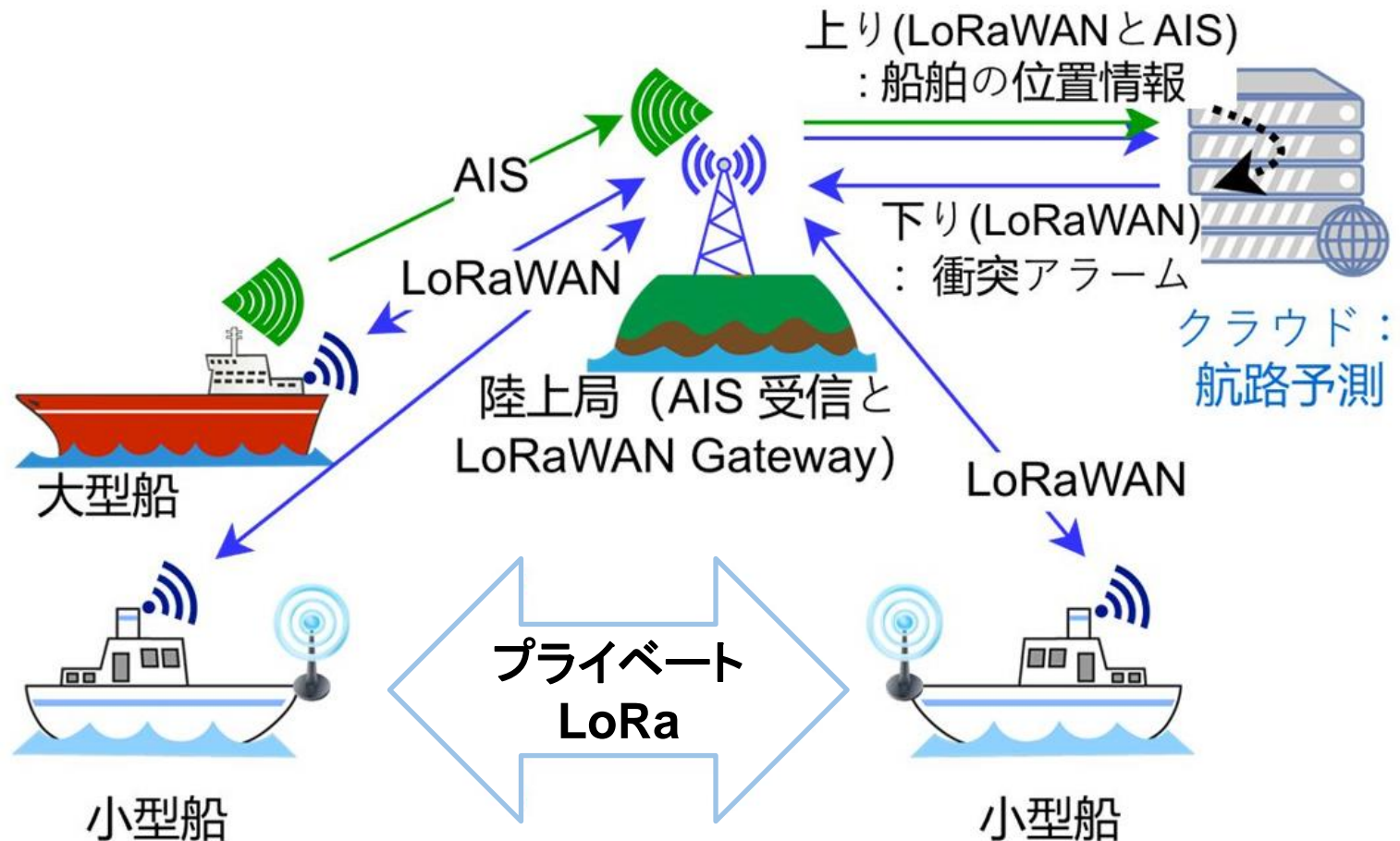


# LoRaWANの概要



TTNでは、ゲートウェイを共有することができる

# 本システムにおけるLoRaWANの利点



LoRaWANではAIS情報も含めて衝突予測し、  
小型船に警告できる

# LoRaWANにおけるDRとペイロードサイズの関係

電波の伝搬距離 ↑長 ↓短	DR	SF	BandWidth /kHz	ビットレート /bps	最大ペイロードサイズ /byte	通信容量 ↑小 ↓大
	0	12	125	250	N/A	
	1	11	125	440	N/A	
	2	10	125	980	11	
	3	9	125	1760	53	
	4	8	125	3125	125	
	5	7	125	5470	222	
	6	7	250	11000	222	
	7	FSK	-	50000	222	



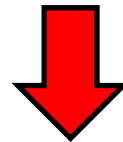
# AIS とは？

**AIS**：大型船に設置が必須の船舶自動識別システム

**簡易AIS**：小型船向けの船舶自動識別システム(必須ではない)

## 簡易AISの問題点

- 周辺の船の位置情報が分かるので漁場情報の流出



漁場情報を秘匿しながら衝突を防止するため、  
**衝突リスク**があるときのみ**警告**する

# IoTサーバーとは？

➡ 本研究室で構築しているデータベースサーバー



データの  
グラフ化, 地図化

## 二次利用

- ・ 商船の物流管理
- ・ 事故船のドライブレコーダー代わり

# 船舶からTTNへ送る GPS情報

取得したGPSデータ

時間 : 18:25:25  
緯度 : 33.84794  
経度 : 132.77052  
海拔 : 125  
速度 : 0  
方角 : 195



2進数変換

88 bit

000011110110010110010110110100101100111010  
0110010010110011111100011111010000000011000011

0:00からの経過分    秒    緯度整数部    緯度小数部  
経度整数部    緯度小数部    海拔    速度    方角



16進数変換

11 byte

0f6596d2ce992cfc7d00c3

# 衝突予測サーバーと エンドデバイスの実行結果

## デバイスからのメッセージ受信

```
---- 2024-02-08 16:04:00 ----
エンドデバイスからのデータを受信
受信データ:
{'device_id': 'eui-70b3d57ed0063d14', 'hour': 16, 'min': 3, 'sec': 59, 'latitude': 33.84789, 'longitude': 132.77072, 'location': {'lat': 33.84789, 'lon': 132.77072}, 'alt': 5, 'speed': 13, 'course': 10, 'RSSI': -85, 'SNR': 12.2, 'RCVdate.UTC': '2024-02-09T07:03:59.191326036Z'}
前回の受信データと比較:
{'eui-a84041d34184d8cf': {'hour': 16, 'min': 3, 'sec': 29, 'lat': 33.8477, 'lon': 132.77074, 'speed': 0, 'course': 1}, 'eui-70b3d57ed0063d14': {'hour': 16, 'min': 2, 'sec': 59, 'lat': 33.84761, 'lon': 132.77139, 'speed': 1, 'course': 24}}
```

## 各デバイスとの衝突予測計算

```
60秒後に50m以内となるデバイスを計算
結果: (デバイスID, そのデバイスの方角, そのデバイスの速度)
[('eui-a84041d34184d8cf', 1, 0), ('eui-70b3d57ed0063d14', 10, 13)]
```

各デバイスへのメッセージ作成

```
time: 1.0 = 0001
course, speed: 10, 13 = 0001010,000001101
警告メッセージ: (...方角7ビット 速度9ビット 時間4ビット) 00010100000011010001
eui-a84041d34184d8cfへメッセージをパブリッシュ: 140D1
衝突が予測されたデバイスデータをESに投入
ES投入完了
```

## デバイスA への警告

```
time: 1.0 = 0001
course, speed: 1, 0 = 0000000010000000
警告メッセージ: (...方角7ビット 速度9ビット 時間4ビット) 00000000100000000001
eui-70b3d57ed0063d14へメッセージをパブリッシュ: 801
衝突が予測されたデバイスデータをESに投入
ES投入完了
```

## デバイスB への警告

## エラスティックサーチへ投入

```
受信データをエラスティックサーチに投入
ES投入完了
受信データをローカルディスクにcsvとして保存
CSV追加完了.
```

```
pi@raspberrypi:~$ cat 2024-02-08.log
2024-02-08 16:04:00,000 - INFO - try SEND
2024-02-08 16:04:10,008 - INFO - UL sented
2024-02-08 16:05:00,000 - INFO - try SEND
2024-02-08 16:05:10,011 - INFO - UL sented
2024-02-08 16:05:20,024 - INFO - DL受信: 140D1
```

## デバイスAの送受信ログ



```
pi@raspberrypi:~$ cat 2024-02-08.log
2024-02-08 16:03:30,000 - INFO - try SEND
2024-02-08 16:03:40,008 - INFO - UL sented
2024-02-08 16:04:30,000 - INFO - try SEND
2024-02-08 16:04:40,011 - INFO - UL sented
2024-02-08 16:04:50,024 - INFO - DL受信: 801
```

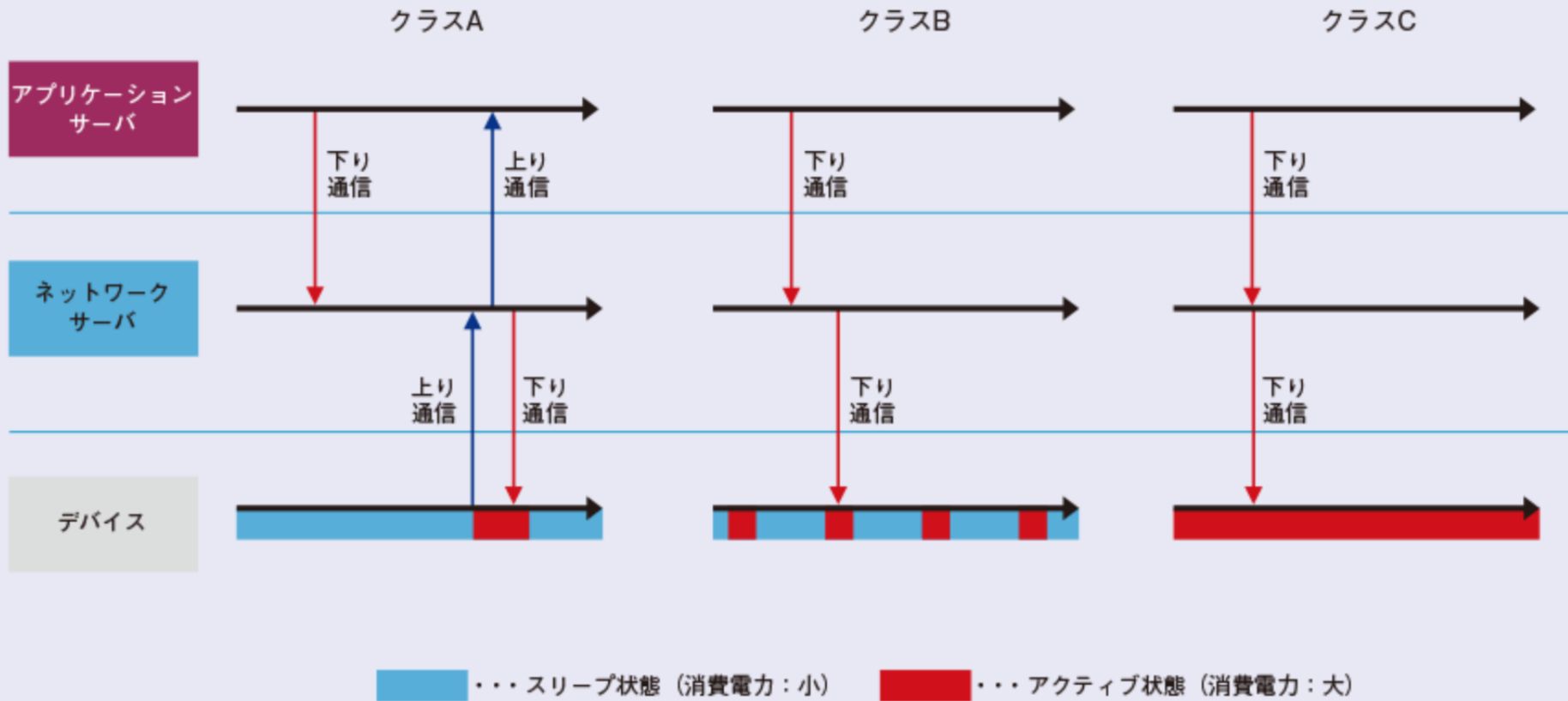
## デバイスBの送受信ログ



## 衝突予測サーバーの実行結果



# LoRaWANのクラス



節電能力 : クラスA > クラスC

下り通信(ダウンリンク)の応答性 : クラスA < クラスC

エンド  
デバイスA



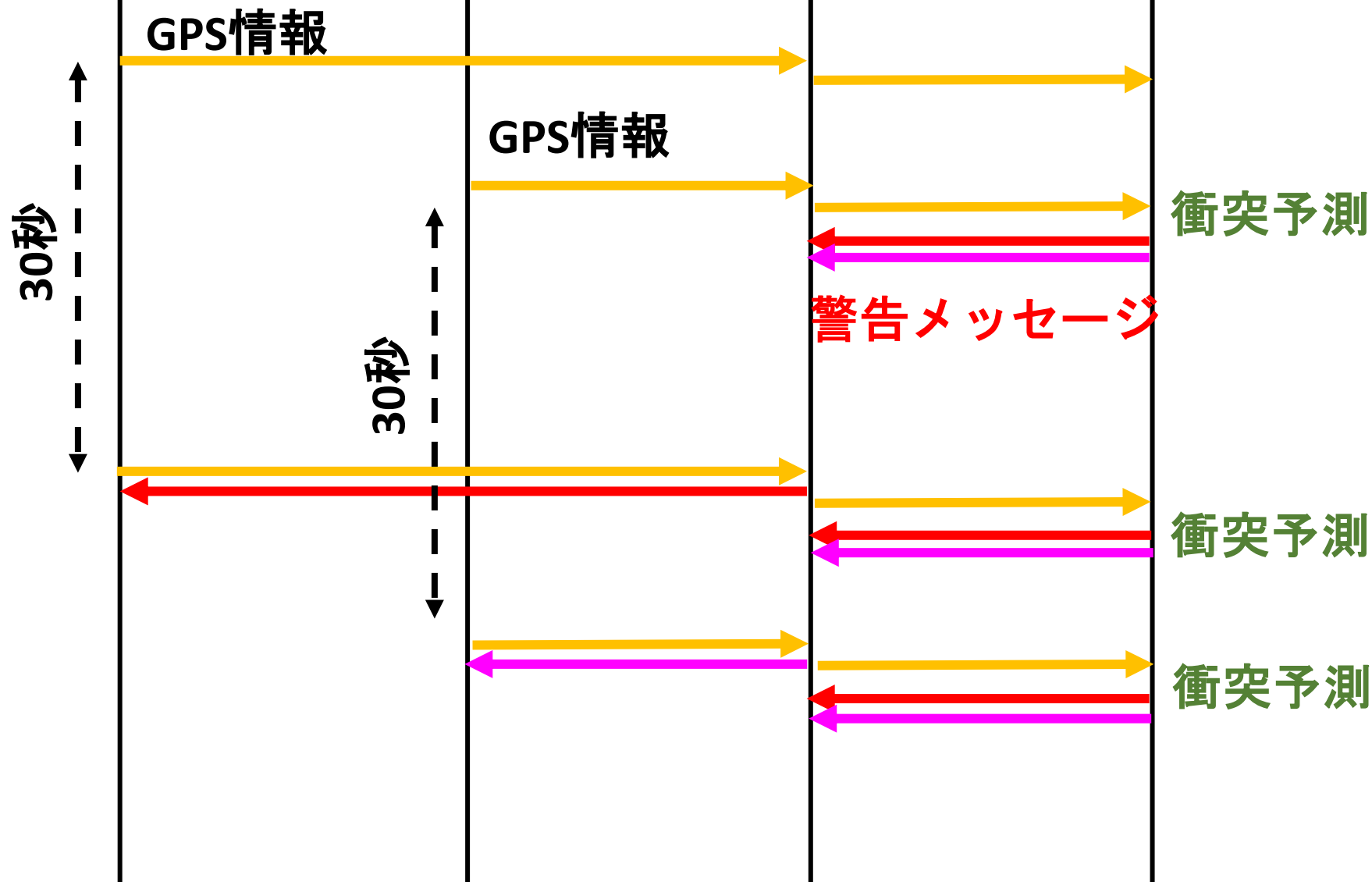
エンド  
デバイスB



TTN  
THE THINGS  
NETWORK



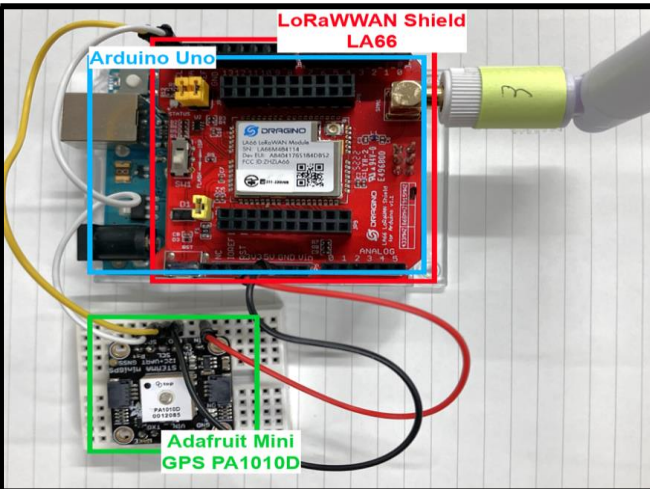
衝突予測  
サーバー



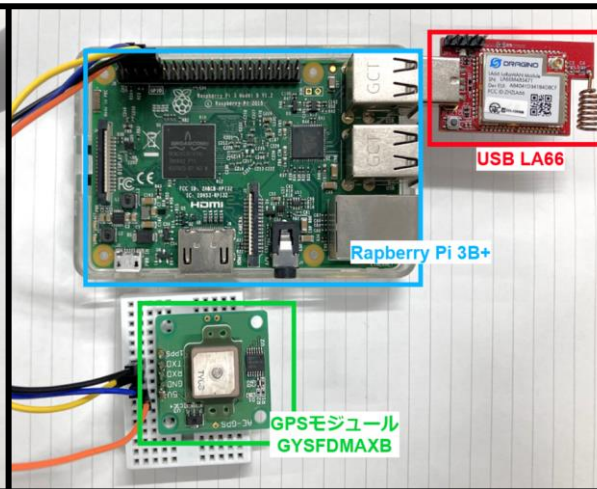


# 3種類のエンドデバイスの作成

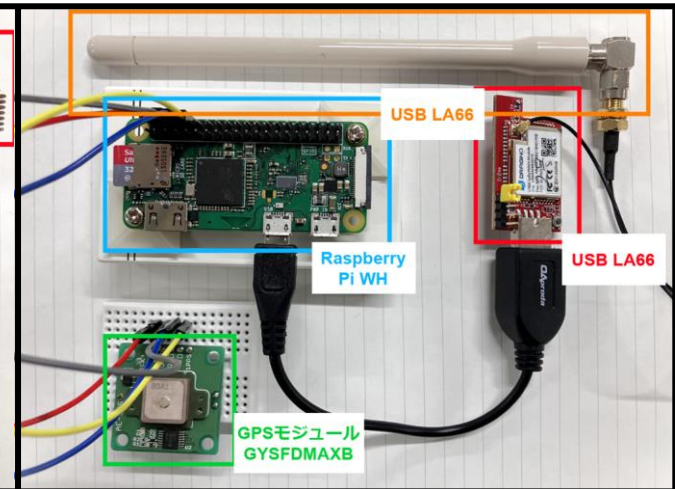
<Arduino LA66>



<USB LA66>



<USB LA66>  
(アンテナ付け替え)



- ・ GPSデータ収集し， LA66(LoRaモジュール)を使って TTNへ送信するプログラムをマイコンに作成

# RSSI測定の測定方法



## ◆ゲートウェイ

高さ : 0.9 m

## ◆エンドデバイス

高さ : 0.9 m

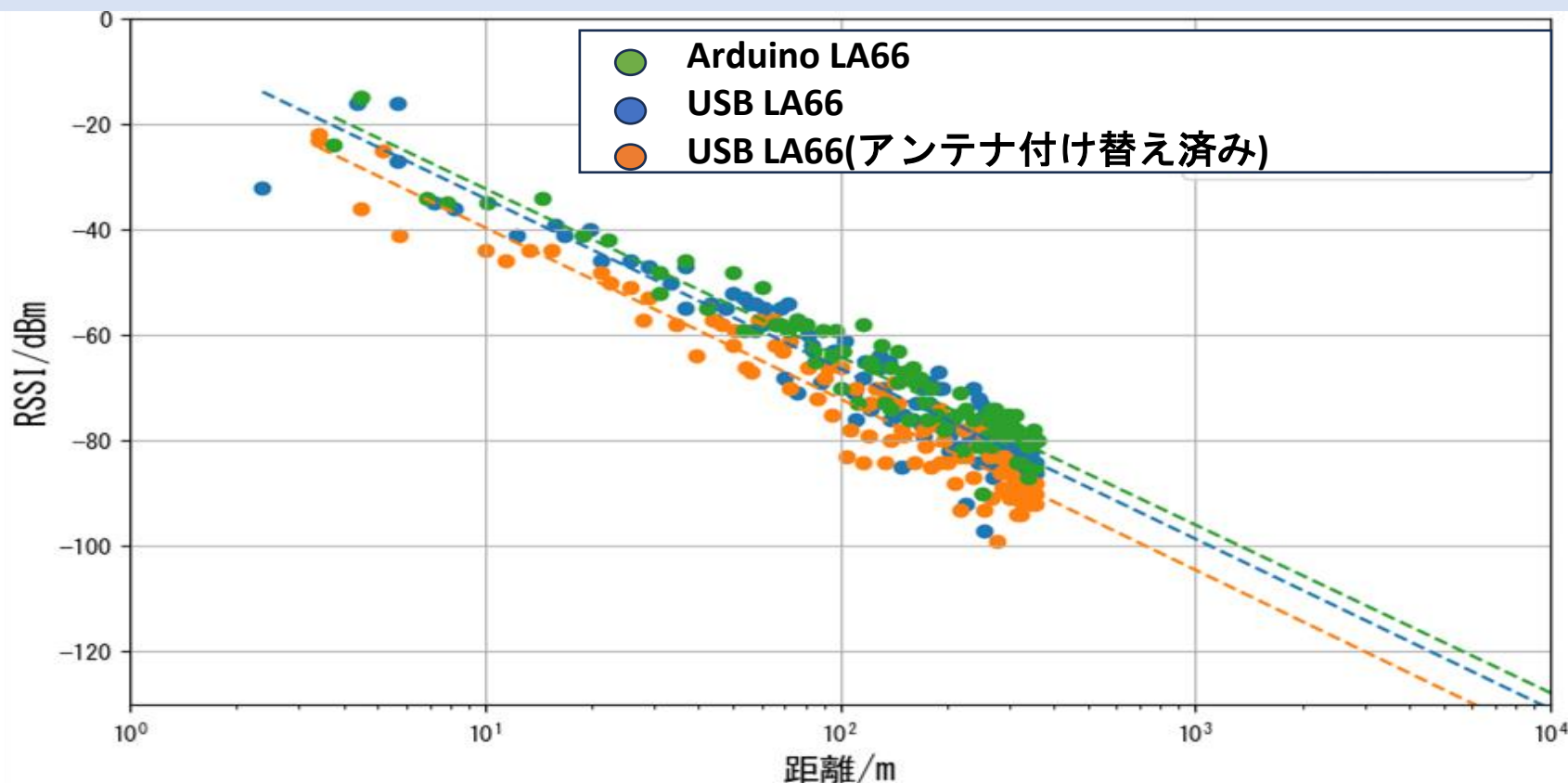
速度 : 0.5 m/s

## ◆最長通信距離

約 350 m

## ◆ゲートウェイを固定し エンドデバイスを移動 させRSSIを測定

# 各エンドデバイスのRSSIの距離特性



通信距離350mまでの距離で、どのエンドデバイスがRSSI特性が良いか測定

# LA66 USB 回路図

初期アンテナ  
取り付け部分

アンテナを取り付ける  
際、マッチング回路を  
外しアンテナを点Aの  
端子口に取り付けた。

