

UAV(ドローン)の案が出たきっかけは、上空にネットワークを構築したいということから。
 「既存の研究は1台のみの実装や、複数台の UAV であってもマルチホップネットワークの研究は理論研究やシミュレーション評価が多く、実装している研究が少ないという課題があるので、私は複数台の UAV を使用し(ZigBee で)実装する。」

活用例) 災害, 点検, 測量, 高速道路, 配送, 農業, ライトショー,

【上空にネットワークを構築することの利点】

- ① 建物の障害を考慮して考えなくて済む.
- ② 地上から高層ビルの最上階の状況を一瞬で詳細に把握できる.
- ③ 航空画像を安価な方法で取得できる.(ZigBee は不可)
 >画像は UAV 自体が保持し, 撮影タイミングや位置だけを指示するデータのみ送受信.
- ④ 自動運転のサポート.(ZigBee では不安?)
 >01 のデータなら, UAV と自動車で送受信できるかも?

【複数台の UAV を無線マルチホップネットワークで接続する利点】

- ① 調査や点検の時間を削減でき、人が立ちいらないので安全に行える。
 >1つの UAV (ドローン) で調査・点検はできるが、1 km²の範囲が 10 個 20 個の場合、とても時間がかかるため、複数の UAV が連携することで、広範囲の調査を一気にすることができる。
- ② 複数の角度を撮影できるよう各 UAV の位置を把握することで、3D 画像を生成できる.
- ③ 広範囲にネットワークを構築できるため、操縦者からの電波が届かない距離にでも UAV を飛行させることができる.
- ④ ライトショーで使用されている無線規格は不明だが、ライトショーでの活用を検討.
- ⑤ 広範囲の海や湖やダム等にも利用可能.

[1]

UAAV (Unmanned Aerial and Aquatic Vehicle) マルチホップネットワークについての研究で、無人航空機（ドローン）や水中ロボットなどの無人機を使用して、データを中継し、ネットワークを実装している。

・ AANET 通信 UAV で構築されたアドホックネットワーク

Wi-Fi として知られる IEEE 802.11(b/n/ac)標準を使用。

IEEE 802.11n メッシュ構成の場合、標準の IEEE 802.11s メッシュ実装が使用される。IEEE 802.11s は、メッシュトポロジ構成をサポートするために、他の IEEE 802.11 の下位層を拡張するしたもの。

IEEE 802.11ac に関しては、UAV ネットワークにおけるこの技術の最初の実装で、屋内では IEEE 802.11n と比較して、より高いデータレートと優れたスループットを示す。屋外では、UAV が基地局から遠ざかると、IEEE802.11ac はスループットの大幅な低下を示すため、AANET での IEEE 802.11ac の使用法に関するさらなる研究の必要性がある。

赤外線技術

利点)低コストの通信システム、

赤外線スペクトル帯域は世界中で規制されていない（これにより国際互換性が可能になる可能性がある）、

欠点)不透明な物体（壁など）を透過できないこと、多くの熱雑音源（太陽光、照明装置など）が存在すること

・ AQNET 通信(この論文で呼ばれている名称)

無人水上車両または無人水中車両で構築されたネットワークのこと。

これにより、水中で取得したデータを無人水上車両(ドローンでも可)で集めて、陸地へ送信できる。

この研究では無人水上車両同士は XBee を使用している。

無人水上車両からは衛星通信ではなく Wi-Fi を使用している。

この論文では、参考文献の紹介が多かったので、その参考文献を今後参考にする。

その他、調べた無償論文は理論研究が多かった。

参考文献

[1] J. Sánchez-García, et al., “A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications”, Computer Communications, Vol 119, April 2018

【研究進捗】

- ① Wireshark に送受信しているデータのみを表示させたいため、随時発信しているビーコンの回数を調整する。
＞まだビーコンのプログラムコードを調査中
- ② Dissector・・・Wireshark のプロトコル解析部分で、バイト列を人が理解できる内容に変換し表示することを可能にする。
 1. [Wireshark の Dissector を使った独自プロトコル解析をやさしく解説してみました \(cyberdefense.jp\)](#)
 2. [\(続\) Wireshark の Dissector を使った独自プロトコル解析をやさしく解説してみました \(cyberdefense.jp\)](#)
 3. [続々 Wireshark の Dissector を使った独自プロトコル解析\(TCP,UDP 分割パケットの場合\) \(cyberdefense.jp\)](#)

以上のサイトを参考にする。

Dissector は Lua 言語を使用し作成する。

-Lua が動くように lua-5.4.6_Win64_bin.zip をダウンロードし環境構築した。

-上記の 1 の URL より Lua のプログラムを Wireshark に取り入れる手順を行ったが、Wireshark に反映されない。