

情報指向ネットワークにおけるネーミングスキーマを用いた 移動可能ルータの実現

高 宇豪[†] 北川 拓[†] 阿多 信吾^{††} オムスーヨン[†] 村田 正幸[†]

[†] 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

^{††} 大阪市立大学大学院工学研究科 〒558-8585 大阪府大阪市住吉区杉本 3-3-138

E-mail: [†]{u-kou,t-kitagawa,suyong,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp

あらまし 近年、ホストではなくコンテンツ主体のネットワークである情報指向ネットワーク (ICN : Information-Centric Networking) は IoT 環境において柔軟な制御が可能なネットワークとして注目されている。我々の研究グループでは、ICN における柔軟な制御に関する研究として、分断ネットワーク間での通信を実現するためのドローンを用いた移動ルータの研究を行ってきた。これまでの研究では、移動ルータの制御はコンテンツリクエストに応えるための受動的な移動制御のみを対象としていた。しかし、ルータの連携を考慮したコンテンツ取得などの戦略を実現するためには、行動戦略や行動に関するアルゴリズムに基づいた能動的な移動制御が必要不可欠である。そこで、本稿では、移動ルータに対し能動的な制御が可能なシステムの実現を目的として、ICN のネーミングスキーマと ICN のストラテジ層を活用した移動制御システムを組み込んだ移動ルータ FR (Flying Router) を提案する。そして、FR を用いた通信の実証のために実機検証を行い、ICN ネーミングスキーマを FR の移動制御 API として使用可能であり、FR の能動的な制御が可能であることを示した。

キーワード 情報指向ネットワーク、移動可能ルータ、センサネットワーク、ネーミングスキーマ、ストラテジ層、移動戦略

Design of Flying Router Control System Based on Name of Packet in Information-Centric Networking

Yuhao GAO[†], Taku KITAGAWA[†], Shingo ATA^{††}, Suyong EUM[†], and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University

1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan

^{††} Graduate School of Engineering, Osaka City University

3-3-138 Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka 558-8585, Japan

E-mail: [†]{u-kou,t-kitagawa,suyong,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp

Abstract Recently, Information-Centric Networking (ICN), which is a content-oriented network rather than host-oriented, is attracting attention as a network capable of flexible communication in IoT environment. As a research on flexible control in ICN, our research group has been researching a movable router using a drone to realize communication in fragmented networks. In previous studies, control of movable routers was targeted only for passive mobility control which occurred when a router receives a content request. However, in order to realize a strategy such as content acquisition in consideration of router cooperation, active mobility control based on behavioral strategies and algorithms related to behavior is indispensable. In order to realize a system that can actively control movable routers, we suggest a movable router FR (Flying Router) into which a mobility control system that utilizing ICN's naming schema and ICN's strategy layer. Through verifying communication between the mobility control system and FR, we show that it is possible to use the ICN naming schema as FR movement control API, and to enable active control of FR.

Key words Information-Centric Networking, Flying Router, sensor network, naming scheme, strategy layer, strategy of routers' moving

1. はじめに

近年、ホストではなくコンテンツ主体のネットワークである情報指向ネットワーク (ICN: Information-Centric Networking) が、IoT 環境における柔軟な通信が可能なネットワークとして注目されている。ICN は TCP/IP とは異なり、通信において機器ごとの IP アドレスの指定が不要であり、多数の機器を扱う IoT 環境やアドレス変更が頻発するモビリティ環境に適している。また、近年では ICN のネーミングスキーマを活用したより柔軟な通信が注目されている。具体的には、コンテンツの名前に優先度の情報を付加してパケット転送方法を柔軟に変更したり、情報の取得以外にもアクチュエーターを動作させたりするといった活用事例が挙げられる [1]。

我々の研究グループでは、ICN における柔軟な制御に関する研究として、センサネットワーク環境や災害時における分断ネットワーク間での通信を実現するためのドローンを用いた移動ルータの研究を行ってきた。これまでの研究では、移動ルータの制御はコンテンツリクエストに応えるための受動的な移動制御のみを対象としていた。しかし、コンテンツの優先度に基づくコンテンツ取得などの戦略を実現するためには、能動的な移動制御が必要不可欠である。移動ルータの移動制御において優先度などの属性を考慮する戦略を適用できれば、従来の手法と比べより柔軟で高速なコンテンツ取得が可能となる。

そこで、本稿では、移動ルータに対し戦略に基づく移動制御が可能なシステムの実現を目的として、ICN のネーミングスキーマと ICN のストラテジ層を活用した移動制御システムを組み込んだ移動ルータ FR (Flying Router) を提案する。ICN の実装としては NDN (Named Data Networking) を対象とし、NDN のパケットフォワーダである NFD (NDN Forwarding Daemon) の拡張によって移動制御システムを実現するために必要な API、制御用コンポーネント、パケット構造を設計した。具体的には、API に関しては、移動経路を設定するパスプランの生成、一覧表示、実行、一時停止、キャンセル、移動速度の設定のための API を設計した。制御用コンポーネントに関しては、FR をユーザインターフェイス、NFD、制御用ストラテジおよびドローン制御プログラムの 4 つの制御用コンポーネントに分割して設計を行った。パケット構造に関しては、Interest パケットを API として用いるために、名前に制御関数の名前を含めた制御用 Interest の名前空間を設計した。以上の ICN におけるネーミングスキーマを活用した FR を用いた通信の実証のために実機検証を行い、ICN ネーミングスキーマを FR の移動制御 API として使用可能であり、FR の能動的な制御が可能であることを示した。

2. 関連研究

近年ではコンテンツベースの柔軟な制御が可能な情報指向ネットワーク ICN に関する研究が活発に行われている。文献 [2] では ICN の Interest パケットの名前に制御命令の名前を付加して送信することで電灯などの IoT 機器の制御手法を提案した。また、文献 [3] では Interest パケットの名前に制御命令の

パラメータを付加して送信することで、その Interest パケットを受信したノードに対してより複雑な制御の実現手法を提案した。パケットの中身ではなくパケットの名前から制御命令を取得できればより簡単に高度な機械制御を実現可能である。

我々の研究チームではドローンにルータを装着してルータの命令に制御されるドローン・ルータに対し、ICN をドローン・ルータのネットワーク層に実装した ICN ルータの FR (Flying Router) に関する研究が行われている [1]。ドローン・ルータのネットワーク層に ICN を実装する理由としては、ICN ではパケット送信時に受信側の IP アドレスを必要としないため、IP アドレスが頻繁に変化するモビリティ環境における通信が TCP/IP より容易に実現できるからである。また、ICN ではパケット受信時にドローンの移動制御などの追加処理を実行できるネットワークレイヤであるストラテジ層を有しており、ストラテジ層を使用すれば簡単にドローン制御を実装できることも理由として考えられる。さらに、ICN パケットのネーミングスキーマに行動戦略の名前や制御命令の名前を付加し、ストラテジ層で様々な場面の行動戦略を定義し、これらのモジュールを用いることでドローン・ルータをより柔軟に制御できると考えられる。

また、文献 [4] では FR を用いた分断ネットワーク通信において、制御システムのシーケンス設計、通信用コンポーネント設計などのプロトコル設計および実機を使用しないシミュレーションの実装を行ったが、実機を用いた実装が行われておらず、その実装は今後の課題になっている。この文献では FR の用途はスマート農業において農作物に取り付けた IoT センサからなるセンサネットワークの通信で FR を使用可能であると提案した。この研究は ICN の柔軟な制御を可能とする構造を活用したアプリケーションの一例として位置付けられている。

3. FR 移動制御システムの設計

3.1 概要

本節では FR 移動制御システムの環境や入出力などのシステム仕様とシステム構造について説明する。最初に、システムの使用環境についてシステムの使用場面、ネットワークトポロジを説明する。次に、システムの仕様についてはシステムの入出力、システム提供する FR 移動制御の内容を述べ、それからシステムの制御を実現するコンポーネントとその役割を紹介する。そして、システムの制御を実現する通信の流れを説明する。

本システムの使用場面としては災害時のネットワークのメッセージ運搬や、スマート農業において農作物に取り付ける IoT センサ間のコンテンツ運搬などの場面は例として挙げられ、その他に FR をパケット通信の中継ノードとして用いたコンテンツ取得などの場面において本システムが使用可能である。ネットワークトポロジに関しては、ユーザノード一台と FR 一台が無線 LAN で接続されており、無線 LAN ではユーザノードがアクセスポイントとなっている。FR は無線 LAN の通信可能範囲内のみ移動可能であり、FR の安全のために通信可能範囲を超える移動が不可能になっている。

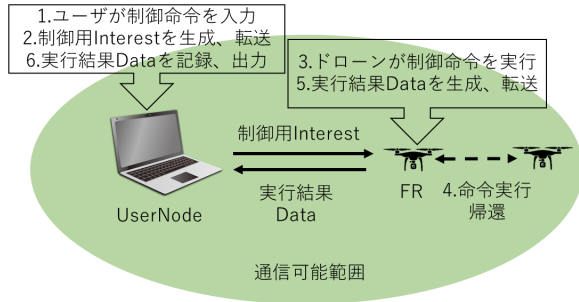


図 1 FR 移動制御システムのネットワークトポロジ

本システムの入力はユーザや他ノードからの制御命令の名前およびそのパラメータを含めたパケット（本稿では制御用 Interest と呼ぶ）であり、その入力のパケットを受信すると FR に対して機器制御或いはメッセージ（本稿では実行結果 Data と呼ぶ）の出力を行う。また、システムが提供する FR 移動制御の内容については、下表ではその制御を実現する API の名前、動作概要、入出力をまとめた。下表の API を実装する目的としては他の利用場面のアプリケーション実装に拡張用のベースの API を提供することである。ただし、下表の API を実現する機材のドローンは離着陸、指定場所への移動、速度調整の API を持たなければならない。

表 1 FR 移動制御を実現する API

API 名	/FRControl/{FRName}/PathCreate/{PathName}/{Point}* Point = /{North}/{East}/{Altitude} {PointName} PointName = {FNName} HOME
動作概要	移動経路の生成、保存、編集
入力	移動経路を指定する際の移動経路の名前の文字列、移動経路の各地点について long 型のユーザノードから地点までの東西方向距離及び南北方向距離及び高さ（単位はメートル）、或いは地点の名前の文字列
出力	なし
API 名	/FRControl/PathList
動作概要	移動経路の一覧表示
入力	なし
出力	保存した移動経路の各地点の FR からの東西方向距離及び南北方向距離、地点の名前
API 名	/FRControl/{FRName}/PathRun/{PathName}
動作概要	経路に沿う移動を開始
入力	移動経路の名前の文字列
出力	なし
API 名	/FRControl/{FRName}/PathPause
動作概要	現在実行中の移動の一時停止と移動の再開
入力	なし
出力	なし

API 名	/FRControl/{FRName}/PathCancel
動作概要	現在実行中の経路に沿う移動をキャンセルし、FR の移動を停止させる
入力	なし
出力	なし
API 名	/FRControl/{FRName}/SetVehicleSpeed/{Speed}
動作概要	FR の移動速度を設定する
入力	機体移動速度（単位は m/s）
出力	なし

本システムのコンポーネントはユーザインターフェイス (UI)、NFD、制御用 Strategy である FRControlStrategy、ドローン制御プログラムである。UI はユーザの入力に基づいて制御用 Interest を生成するプログラムである。その役割としてはネットワーク接続などの初期設定、制御命令を格納したパケットの生成および転送、実行結果の出力を行っている。NFD は NDN パケットを転送するプログラムである。NFD は NDN パケットの転送、ノードの名前などの変数の保持を担当している。また、NFD の中では FR の移動制御のための関数が定義されている制御用 Strategy を有している。その役割としては制御用 Interest の内容に基づいてドローン制御プログラムの関数の呼び出しを行っている。ドローン制御プログラムはドローンの実際の動作を記述したプログラムである。ドローン制御プログラムは実際のドローンの移動制御を行っている。

制御を実現するためのシステム内の通信の流れについては、最初にユーザは制御命令などの制御用情報を UI に入力し、UI は入力の情報を制御用 Interest の名前に格納し、制御用 Interest を生成して指定の FR に転送する。制御用 Interest が指定の FR に到着すると、FR はその制御用 Interest の名前に基いてドローン制御プログラムの関数を呼び出す。ドローン制御プログラムの関数はドローンに制御命令を転送して FR の移動制御を行う。そして、移動制御が終了した後、ドローン制御プログラムの関数の実行も完了となり、実行結果を格納した Data パケットを生成してユーザノードへ転送する。Data パケットを受け取ったユーザインターフェイスはそのパケットの内容を標準出力に出力する。

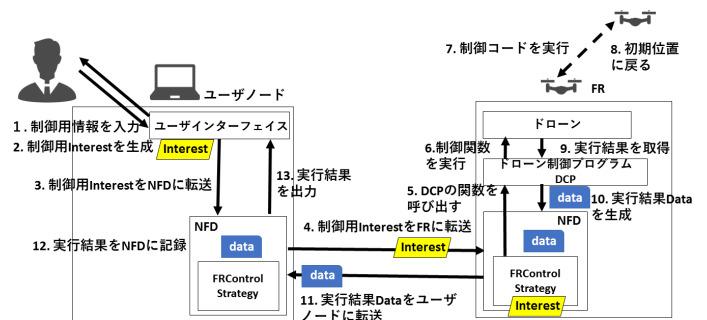


図 2 FR 移動制御システムのコンポーネントダイアグラム

3.2 FR 移動制御システムの packets 構造とコンポーネントの設計

本節ではコンポーネント間通信の packets 構造および各コンポーネントの仕様として入出力、例外処理、使用ツール、およびコンポーネントの実装方法について説明する。packets 構造については制御用 Interest と実行結果 Data の名前空間を説明する。

3.2.1 packets 構造

制御用 Interest は FR の移動制御の実行を指示するための ICN packets である。制御用 Interest は /FRControl の prefix を持ち、名前以外の packets 構造は ICN の Interest packets と同じである。制御用 Interest を受信すると制御用 Strategy 内の移動制御を実現する関数を呼び出し、移動制御が終了したら Interest は InterestLifeTime 後に自動的に削除される。Interest packets の名前空間は /FRControl/{FRName}/{Function}/{Parameter} の書式を用いており、FRControl は制御用 Interest であることを判断する prefix であり、FRName は制御する FR の名前であり、Function は移動制御の API の名前であり、Parameter は API のパラメータである。

/FRControl の prefix を持つ Interest の中で以上の書式ではないものは命名規則に従わない packets と見なされ破棄される。また、/FRControl の prefix を持たない Interest はコンテンツを要求する Interest としてみなし、制御用 Strategy の処理を呼び出さずに NFD のフォワーディングのアルゴリズムに従って処理される。下図では制御用 Interest の名前空間を示している。

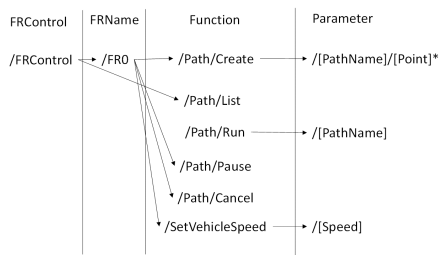


図 3 制御用 Interest の名前空間

また、実行結果 Data は FR 移動制御の API の実行後の戻り値を格納した ICN packets である。実行結果 Data は ICN の Data packets と同じ構造を持ち、名前空間に関しては実行結果 Data は制御用 Interest と一対一の関係であるためと名前空間は同じであり、格納したコンテンツは API の戻り値である。本システムでは API の戻り値を利用されていないため、実行結果 Data の生成を行わないようにした。

3.2.2 ユーザインターフェイス (UI)

UI は API の入力受付および制御用 Interest の生成と転送を行う NFD の拡張プログラムである。UI はユーザノードにインストールしている。ユーザインターフェイスを起動すると FR を制御するための初期設定は自動的に行われ、FR 移動制御システムが起動して入力を受付を開始する。また、UI の仕様に関

しては入力ユーザノードの標準入力からの API の名前とそのパラメータの文字列であり、出力はユーザノードの NFD へ制御用 Interest を出力する。例外処理としては 3.2.1 節の名前空間以外の文字列を入力された際に入力を無視して次の入力を求めることである。

また、UI の主要動作としては、まず、ユーザノード NFD を起動して制御用 Interest 転送先を設定する。次に、FR に遠隔ログインし、FR 側で NFD を起動して制御用 Strategy を適用する。そして、UI はユーザの標準入力を受け付け、入力を確認したら制御用 Interest を生成して NFD に転送し、次の入力を求める。これらの動作の実装方法に関しては、NFD の起動、制御用 Interest 転送先の設定、制御用 Strategy の適用は NFD の API で実装しており、制御用 Interest の生成と転送は NDN のライブラリである ndn-cxx の API で実現可能である。

3.2.3 NFD と制御用 Strategy

NFD は packets フォワーディング、テーブルなどの変数の保存と取得、ストラテジ層の処理の呼び出しを行うプログラムである。NFD はユーザノードと FR にインストールしている。本システムでは NFD にノードの名前、移動経路のリスト、他ノードの名前と座標のマップおよび制御用 Strategy を追加しており、その他に使用している NFD の機能としては packets の転送であり、その機能は拡張を行わずに使用している。

また、制御用 Strategy は制御用 Interest に基づいてドローン制御プログラムを呼び出す NFD の拡張ストラテジである。制御用 Strategy はユーザノードと FR 両方にインストールしている。制御用 Strategy の仕様に関しては入力は NFD からの制御用 Interest であり、入力を確認したら制御用 Interest に基づいてドローン制御プログラム内のドローンの移動を制御する関数を呼び出す。例外処理としては制御用 Interest の制御命令の名前とパラメータの文字列は定義されていない形の場合、制御用 Interest を破棄する。

制御用 Strategy の主要動作については、制御用 Interest の名前の制御命令の名前とそのパラメータの文字列を抽出し、その文字列に基づいてドローン制御プログラム内の関数を呼び出している。その実装方法に関しては、制御命令の名前とそのパラメータの文字列の抽出は ndn-cxx の API で行い、ドローン制御プログラム内の関数の呼び出しはコマンドラインでドローン制御プログラムにオプションを渡すことで関数の指定を行う。

3.2.4 ドローン制御プログラム

ドローン制御プログラムとは実際にドローンの移動制御を行う拡張プログラムである。ドローン制御プログラムは FR にインストールしている。ドローン制御プログラムの仕様としては入力は制御用 Strategy からコマンドライン引数を受け取り、入力を確認したらコマンドライン引数の指定したドローン移動制御の関数を呼び出す。例外処理としてはドローンと通信できない場合にプログラムが終了する。

ドローン制御プログラムの主要動作としてはドローンの移動制御の関数の実行であり、その実装方法としてはドローン制御ツールの Dronekit-Python が提供するドローン制御用 API の組み合わせでドローンの離着陸、指定地点への移動などを実装

する。

3.3 FR 移動制御システムの API のシーケンス設計

本システムの API のシーケンスは呼び出す関数の名前が異なるだけでシーケンスの呼び出し関係は同じであるため、本節では API のシーケンスの例として PathRun のシーケンスについて説明する。PathRun のシーケンスでは最初にユーザが UI に制御命令とそのパラメータを標準入力から入力し、UI はその情報を制御用 Interest の名前に付加し、制御用 Interest を生成してユーザノードの NFD へ転送する。次に、制御用 Interest を受け取った NFD はそれを制御用 Strategy に渡してノードの名前を確認し、制御対象は FR であるためここでは制御用 Strategy の関数を呼び出さずに NFD の処理に戻る。そして、ユーザノードの NFD は制御用 Interest を FR の NFD へ転送し、それを受け取った FR の NFD は同様に制御用 Strategy に制御用 Interest を渡してノードの名前を確認する。現在のノードは制御用 Interest が指定した FR であるため、制御用 Strategy はドローン制御プログラムの PathRun 関数を呼び出し、ドローン制御プログラムは PathRun 関数を実行してドローンに命令を転送し、ドローンは命令通りに制御される。以上のシーケンスは図 4 で示している。

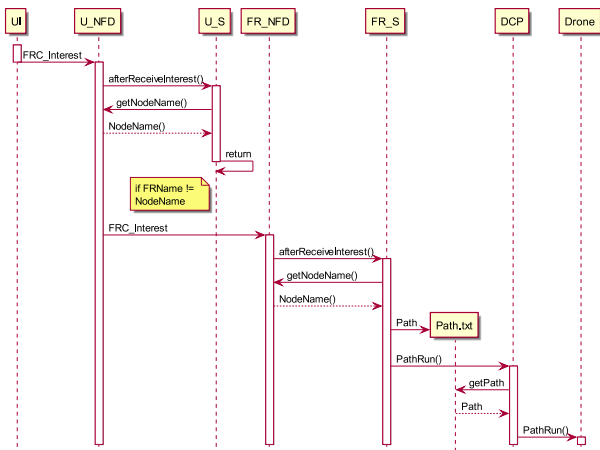


図 4 API の PathRun のシーケンス図

4. 実機検証

本章では FR 移動制御システムの API の動作を確認することで、本システムを用いることで移動ルータに対して能動的制御が可能であることを示す。最初に実装環境、使用機材、実験環境を述べる。次に、システムの検証方法として実験の操作と手順について説明する。そして、検証の結果を示し、ICN のネーミングスキーマは移動制御の API として使用可能であること、および本システムを用いることで移動ルータに対して能動的制御が可能であることを示す。

4.1 実装環境、使用機材、実験環境

表 2 と表 3 では FR 移動制御システムの実装環境に関して使用したソフトウェア、ツール、機材を列挙する。

表 2 実装環境の説明

ツール	ツールバージョン	説明
ndn-cxx	0.5.0	NDN のライブラリ
NFD	0.5.0	NDN パケットフォワーダ
DroneKit-Python	2.0	ドローンの制御ツール
Mission Planner	1.3.44	動作確認で使用し、ドローンの状態を GUI で表示するツール
Python	2.7.0	ドローンの制御ツールの言語
c++	C++11	ndn-cxx と NFD とユーザインターフェースの言語
Ubuntu	14.04 LTS	ユーザノード OS、NFD が正常に動作
Raspbian	Sep 2016	FR の RPi の OS、NFD が正常に動作

表 3 使用機材の説明

機材	機材型番	説明
ドローン	3D Robotics Solo	プログラムによる制御可能、WiFi による通信可能、GPS 搭載
小型コンピュータ	Raspberry Pi 3 Model B	WiFi 搭載、ドローンから電力を供給



図 5 FR の全体図

ドローンと Raspberry Pi (RPi) は USB ケーブルによる通信を行わず、ドローンからシリアルポートによる電力供給のみを行う。その理由は 3DR Solo の仕様によりドローンから RPi を USB ホスト及び USB デバイスとして識別できなかったからである。ただし、ドローンと RPi の通信はドローンのコントローラに付属している無線ネットワークを通じて通信を行う。また、実機検証の場所は大阪大学吹田キャンパス工学部グラウンドの中で高さ 30 メートル、辺の長さ 20 メートルの正方形地域の場所を実験場所とする。

4.2 実験方法

実験対象の API は表 4 の移動経路を生成する PathCreate および PathRun と PathPause を組み合わせて作成した FN-PathRun の二つの API である。FNPathRun は他ノードとのパケット通信の場面を想定して作成した実機実験用の API であり、指定地点への移動および 5 秒の滞在を行う API である。その組み合わせの方法についてはドローン制御プログラムにおいて指定地点への移動の関数と移動を一時中止する関数の組み合

わせのみを行い、その他のコンポーネントの拡張を行っていない。そのため、この API のシーケンスは図 4 のシーケンスと比べ、呼び出す関数の名前のみ異なりコンポーネントの呼び出し関係は同じである。なお、FNPathRun の名前空間は PathRun と同じである。

そして、実験内容については、FR は辺長約 10m の正三角形の経路に沿って移動し、三角形の頂点でそれぞれ 5 秒滞在し、最初の離陸した場所に着陸する。その実現のためにまず API の PathCreate で移動経路を生成し、その後に API の FNPathRun で生成した経路に沿って移動を開始する。確認する項目としては FR の移動などの動作、制御用 Interest が送信されたことを示すデバッグメッセージ、PathCreate の移動経路を生成した後で移動経路を標準出力に表示するメッセージ、FNPathRun の実行中において地点までの距離メッセージと滞在時間メッセージを確認する。

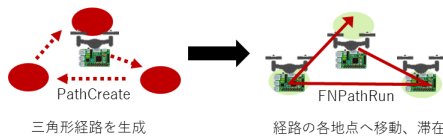


図 6 PathCreate と FNPathRun に関する実験内容

実験手順については、まず、UI を起動して /FRControl/FR0/PathCreate/Path0/10/0/5/-5/10/5/-5/-10/5 を入力する。入力後に標準入力から経路の各地点の座標の出力メッセージを確認する。これにより、Path0 の名前の経路が生成され、その経路の地点の座標は (ユーザノードから北方向の距離、ユーザノードから東方向の距離、高度) の集合で表すと (10, 0, 5), (-5, 10, 5), (-5, -10, 5) の三つの地点を持つ経路が生成される。次に、UI に /FRControl/FR0/FNPathRun/Path0 を入力する。入力後に FR の三つの地点への移動と滞在、最初の地点への帰還を確認し、FR のコマンドラインから地点までの距離のメッセージおよび地点上空での滞在時間のメッセージの出力を確認する。実際の実験では以上の動作確認を行い設計仕様通りの動作とメッセージを確認し、ICN ネーミングスキーマを FR 移動制御 API としての使用が可能であること、および本システムを用いることで FR の能動的制御が可能であることを示した。

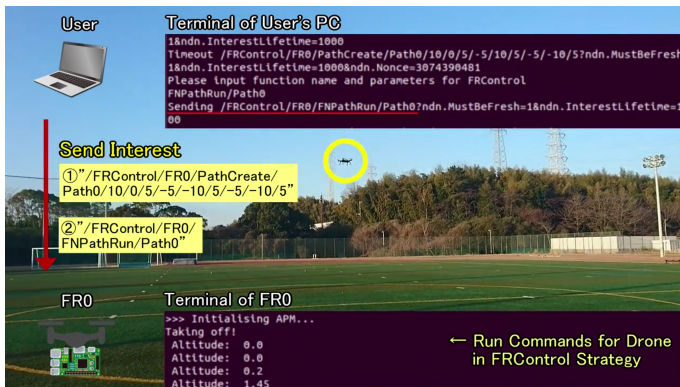


図 7 FR 移動制御システムの指定経路に沿う移動と滞在を実現する API の実機検証

5. おわりに

本稿では ICN のネーミングスキーマとストラテジ層の柔軟性に着目し、移動ルータの能動的制御が可能な FR 移動制御システムを提案した。まず、FR 移動制御システムの提供する API 及びその動作、実装方法を説明した。次に、システムの API の実装のために、移動制御システムのパケットの仕様、および各コンポーネントの仕様と実装方法について説明した。そして、実機検証では最初に使用ツールや OS などの実装環境、使用機材および実験環境について紹介した。次に、主要な API の実験内容、検証手順を説明し、実験の結果としては各 API の仕様通りの動作を確認した。実機検証により、ICN ネーミングスキーマを FR 移動制御 API の使用が可能であること、および本システムを用いることで FR に対する能動的制御が可能であることを示した。

今後の課題としては、まずは FR 移動制御システムに通信機構を実装し、一台の FR を用いたセンサネットワーク通信を実現する。次に、FR 移動制御システムを複数台の FR に導入し、複数台の FR によるセンサネットワーク通信を実現する。また、センサネットワーク通信以外の事例に関して FR 移動制御システムの拡張を行う。

謝 辞

本研究の一部は、独立行政法人情報通信研究機構 (NICT) 委託研究「欧州との連携による情報指向ネットワーキングに関する実証的研究開発」の成果である。本研究に貢献して頂いた先生方、スタッフ方及び学生に厚くお礼申し上げます。

文 献

- [1] T. Kitagawa, S. Ata, and M. Murata, "Retrieving information with autonomously-flying routers in information-centric network," in *Proceedings of 2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.
- [2] W. Shang, A. Bannis, and T. Liang, "Named data networking of things," in *Proceedings of 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2016, pp. 117–128.
- [3] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice-over Content-Centric Networks," in *Proceedings of the 2009 Workshop on Re-architecting the Internet*, ser. ReArch '09, Dec. 2009, pp. 1–6.
- [4] T. Kitagawa, "Mobility-controlled flying routers for information centric networking," Master's thesis, Osaka University, Feb. 2017.
- [5] R. Sugihara and R. K. Gupta, "Path planning of data mules in sensor networks," *ACM Trans. Sen. Netw.*, vol. 8, no. 1, pp. 1:1–1:27, Aug. 2011.
- [6] B. Li, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for icn naming scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, Apr. 2016.
- [7] J. S. Liu, S. Y. Wu, and K. M. Chiu, "Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm," in *Proceedings of 2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, April 2013, pp. 30–37.