

特別研究報告

題目

情報指向ネットワークにおける
自律移動可能なルータを用いた情報取得

指導教員

村田 正幸 教授

報告者

北川 拓

平成 27 年 2 月 13 日

大阪大学 基礎工学部 情報科学科

情報指向ネットワークにおける
自律移動可能なルータを用いた情報取得

北川 拓

内容梗概

近年、IP ネットワークに代わる将来ネットワークの一つとして情報指向ネットワーク (Information Centric Networking : ICN あるいは Content Centric Networking : CCN) の研究が盛んに行われている。CCN は IP ネットワークで行われていたホスト指向の通信とは異なり、経路制御にコンテンツの名前自身を用いることを設計思想としたネットワークアーキテクチャである。情報指向ネットワークアーキテクチャは経路情報として用いる識別子 (名前) の柔軟性の高さを利用し、よりフレキシブルな経路制御によるコンテンツ取得を実現することができる。このような背景から、近年 CCN を利用した柔軟なコンテンツ取得に対する研究が行われている。ここで柔軟なコンテンツ取得とは、名前を指定した静的なコンテンツの取得だけでなく、中継あるいはエンドノードにおけるさまざまな制御も考慮に入れた動的なコンテンツ取得を指す。例えば、名前を直接セッションのシグナリングに利用する方法や、動画取得におけるフレームレート・解像度などを名前に含めることで、経路制御およびノード処理をシームレスに行うことを含めた柔軟なコンテンツ取得が可能となる。しかしながら、既存研究ではエンドノードおよび中継ルータにおけるデータ処理に関する制御が主として考えられているのみであり、機器の物理的な動作、特に実移動を伴う制御についてはあまり検討されていない。本報告では、中継ルータの物理的な移動を含めた経路制御を情報指向ネットワークに組み入れることを考える。コンテンツ名の指定だけで中継ルータの物理移動を含めた経路制御が可能になることで、相互に接続されていない独立ネットワーク (分断ネットワーク) 間での情報流通および共有が実現できる。以上の背景のもと、本報告ではルータの物理的な移動制御を実現するために、CCN ルータを搭載した UAV (Unmanned Air Vehicle) を用いる。そして UAV の移動を情報指向ネットワークの経路制御において行うことによって、取得不可能な遠隔地のネットワークにあるコンテンツを、コンテンツ名を指定するだけで UAV を介して取得可能とする新しい情報指向ネットワークアーキテクチャを提案する。さらに提案アーキテクチャを実現するために CCN ルータと UAV を組み合わせた飛行ルータ (Aerial Router : AR) の設計および開発を行い、分断ネットワークにおけ

るコンテンツ取得のための経路情報の作成，交換手法などについて設計する．また，飛行ルータのプロトタイプ製作を行い，簡便な分断ネットワークによる基礎実験を行うことで提案方式の有効性について検証する．

主な用語

情報指向ネットワーク（CCN），ルーティング，UAV，自律制御，分断ネットワーク

目次

1	はじめに	6
2	関連研究	10
2.1	コンテンツ名に制御情報を含めた CCN アプリケーションに関する研究	10
2.2	分断ネットワークにおける情報流通基盤に関する研究	10
3	ルータの実移動を伴う情報指向ネットワークの提案	13
3.1	提案ネットワークの概要	13
3.2	提案ネットワークにおける経路情報	13
3.2.1	経路情報の定義	13
3.2.2	経路情報の生成方法	14
3.3	提案ネットワークにおけるコンテンツの取得	16
4	提案ネットワークの設計および実装	19
4.1	飛行ルータの設計	19
4.1.1	仮想プロキシ方式の設計	19
4.1.2	巡回モードにおける動作設計	20
4.1.3	配送モードにおける動作設計	20
4.2	独立ネットワークにおけるゲートウェイの設計	21
4.3	提案ネットワークの実装	21
4.3.1	飛行ルータの実装構成	22
4.3.2	仮想プロキシの実装	23
4.3.3	自律移動制御の実装	25
5	実験ネットワークにおける動作検証	29
5.1	実験環境	29
5.2	実験方法	30
5.3	実験結果	32
6	おわりに	35
	謝辞	36
	参考文献	37

目 次

1	CCN における通信フローの例	7
2	VoCCN における通信プロトコル (文献 [1] の Figure 4 から引用)	11
3	Dash over CCN におけるセグメント毎のコンテンツ名 (文献 [2] の Figure 3 から引用)	11
4	CCN Homenet におけるポリシーベースルーティングに拡張された FIB (文献 [3] の Fig. 4 から引用)	12
5	提案ネットワークが想定する環境	14
6	飛行ルータにおける巡回モードと配送モードの動作概要	15
7	巡回モードにおける経路情報の生成例	16
8	配送モードにおけるコンテンツ取得例	17
9	仮想プロキシ方式の動作概要	19
10	GW Proxy の動作概要	22
11	飛行ルータの構成図	24
12	飛行ルータの外観	25
13	Proxy クラスとその関連クラスのクラス図 (一部)	27
14	自律移動制御におけるフローチャート (概要)	28
15	検証実験に用いる地点間ネットワーク	30
16	実験ネットワークにおけるインタレストパケットのルーティング	31
17	実験ネットワークにおけるデータパケットのルーティング	31
18	Node A から Node B に ping コマンドを実行した結果	33
19	飛行ルータが Region A に戻ってきた時の Node A における出力画面	34

表 目 次

1	AR.Drone 2.0 Power Edition の仕様	23
2	Raspberry Pi Model B+の仕様	23
3	飛行ルータのハードウェア仕様	26
4	飛行ルータのソフトウェア仕様	26
5	Node A の FIB	29
6	Node B の FIB	29
7	飛行ルータの FIB	30
8	検証実験で用いる各種定数の値	32

1 はじめに

インターネット黎明期では、電子メールや遠隔操作など通信相手を指定して通信を行うホスト指向型通信が主な利用形態であり、IP (Internet Protocol) では通信相手のノード識別子である IP アドレスを指定した通信が行われている。一方現在のインターネットの利用形態は、コンテンツあるいはサービス指向型へと変化してきており、例えばウェブ利用者は検索エンジンを利用して目的のコンテンツを保持しているサーバを検索し、該当するサーバと通信することで目的のコンテンツを取得する。ここでユーザの興味対象はコンテンツ自身であり、実際のコンテンツがどのサーバにあったかはほとんどの場合関心がない。現在のインターネットにおいては、このように利用形態と実際の通信形態の間に乖離が生じており、その結果コンテンツ流通などにおいて非効率性が生じている。

情報指向ネットワークは、そのような乖離を解消し、コンテンツ指向の通信形態を実現する新しいネットワークアーキテクチャとして、ポストインターネットの研究において注目されている。情報指向ネットワークでは、目的のコンテンツの取得を行いたい場合、そのコンテンツ名自身をパケットの宛先アドレスとして用い、コンテンツ名による経路制御を行うことでコンテンツ指向型の通信形態を実現する。コンテンツ名による経路制御は、コンテンツとその保有ノードの結合性を解消し、コンテンツの独立性および可搬性の向上、さらに任意の位置でのコンテンツキャッシュを可能とすることによる通信性能の大幅な効率化が期待されており、現在多くの研究が遂行されている。

情報指向ネットワークの研究プロジェクトとしては欧州発の PURSUIT [4] や米国発の CCN [5] 及びその派生の NDN (Named Data Networking) [6] 等が有名であるが、本報告で扱う情報指向ネットワークは後者の CCN/NDN に準拠するものとする。CCN では、経路制御識別子としてコンテンツの名前を含むインタレストパケットを生成し、コンテンツ要求者（すなわちクライアント側）から通信を開始する。インタレストパケットを受信した各 CCN ノードは次の手順でルーティング処理を行う。

- (i) インタレストパケットを受信した CCN ノードはまず、PIT (Pending Interest Table) と呼ばれるテーブルにインタレストパケットが送信されてきたインターフェイスを記録する。PIT とはあるコンテンツ名を持つインタレストパケットがどのインターフェイスから送信されてきたか記録しておくテーブルであり、同一の名前を持つコンテンツが返送されてきた際にそのインターフェイス宛に受信したコンテンツを転送する。このような処理を行うことにより、結果的にインタレストパケットが通ってきた経路を逆に辿るような形でクライアントに要求コンテンツを届けることができる。
- (ii) PIT に対するエントリの登録が終わると、CCN ノードは CS (Content Store) と呼ばれるキャッシュ領域を参照し、受信したインタレストパケットに対応するコンテンツ

が自ノードにキャッシュされていないか確認する。コンテンツがキャッシュされている場合はCS内にあるデータパケットをPITに登録されているインターフェイスに返送する。CCNでは中継ノードにてコンテンツのキャッシュを行うことで、コンテンツ取得における応答時間の短縮やネットワークの輻輳回避、コンテンツの可用性向上等を期待することができる。

- (iii) 受信したインタレストパケットに対応するデータパケットがCSに存在しない場合、CCNノードはFIB (Forwarding Information Base) と呼ばれるルーティングテーブルを参照し、次ホップ (インタレストパケットの転送先となるインターフェイス) を検索する。FIBは経路制御識別子としてコンテンツ名 (あるいはそのプレフィックス) を使用することを除き、IPネットワークにおけるルーティングテーブルと同等の役割を果たす経路表である。

上記の手順によるルーティングを各ノードが行うことで、インタレストパケットは最終的にコンテンツを保持するノードに到達する。図1は複数のクライアントが同一のコンテンツ「/fruit/apple.jpg」を要求するCCNの通信フローである。以下にその詳細を述べる。

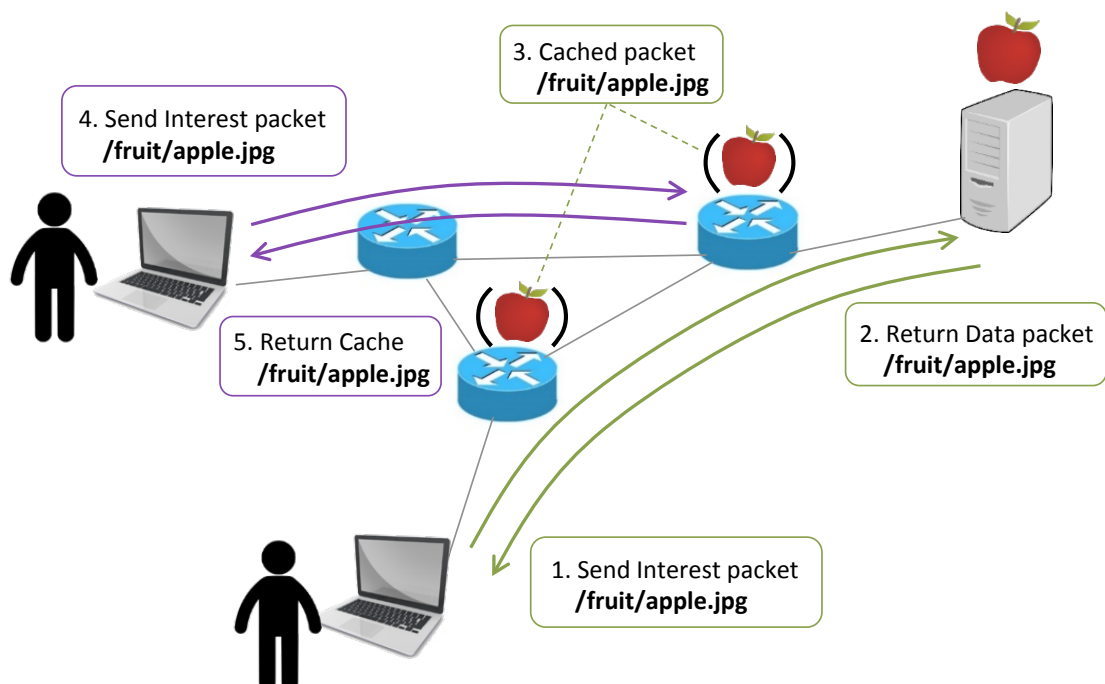


図 1: CCN における通信フローの例

1. クライアントはコンテンツ名として「/fruit/apple.jpg」を含むインタレストパケット

を生成し、FIB を参照することで適切なノードに転送する.

2. インタレストパケットが中継ノードを経て目的のコンテンツを所持するサーバ (目的ノード) に到着すると、目的ノードはコンテンツ名として「/fruit/apple.jpg」を含むデータパケットを生成し、自身の PIT を参照することでインタレストパケットを送信してきたノードに対しデータパケットを返送する.
3. データパケットを受信した各ノードは必要に応じてそのデータパケットを CS にキャッシュする.
4. 別のクライアントがコンテンツ名として「/fruit/apple.jpg」を含むインタレストパケットをネットワークに転送する.
5. インタレストパケットを受信した中継ノードにおいて、コンテンツ名が「/fruit/apple.jpg」であるデータパケットが CS にキャッシュされている場合、インタレストパケットは目的ノードまで到達することなく中継ノードにてキャッシュされたコンテンツが返送される.

CCN のルーティングに関する特徴は上記の通りであるが、さらに近年ではコンテンツの「名前」という経路制御識別子としての高い柔軟性を利用し、Voice over CCN (VoCCN) [1] や Dash over CCN [2] など、コンテンツ名にコンテンツへの制御情報を含めた柔軟なコンテンツ取得を行う研究が行われている. 文献 [1] では VoIP (Voice over IP) に CCN を適用したアプリケーションを取り扱っており、SIP (Session Initiation Protocol) 相当のシグナリングを CCN の名前として表現したパケット交換で行う手法が提案されている. また文献 [2] ではストリーミング映像のビットレートや解像度をコンテンツの名前に含めることでセグメント単位の画像品質制御を実現している.

このように、情報指向ネットワークでは経路制御およびノード処理をシームレスに行うことを含めた柔軟なコンテンツ取得を提供することが可能となるが、既存研究ではエンドノードおよび中継ルータにおけるデータ処理に関する制御が主として考えられているのみであり、機器の物理的な動作、特に実移動を伴う制御についてはあまり検討されていない. 本報告では、中継ルータの物理的な移動を含めた経路制御を情報指向ネットワークに組み入れることを考える. コンテンツ名の指定だけで中継ルータの物理移動を含めた経路制御が可能になることで、相互に接続されていない独立ネットワーク間での情報流通および共有が実現できる.

一方、分断ネットワークにおける通信に関して、例えば災害ネットワーク等における通信を提供する技術に DTN (Delay-Tolerant Network) がある. DTN の既存研究には文献 [7]

や文献 [8], また DTN にコンテンツ指向を適用した ICDTN (Information-Centric Delay-Tolerant Network) [9] 等の研究がなされているが, DTN は本質的に経路制御を主体的に取り組んでおらず, 経路およびコンテンツ転送の最適化について統合的に取り扱うことができない. 一方, 本報告で提案するルータの実移動を伴う情報指向ネットワークではルータによる自律的な移動制御および経路・コンテンツ転送の最適化を行うことができるため, 分断ネットワークにおけるメッセージ到達率・到達時間といった通信性能を向上させることが可能となる.

2 関連研究

本章では、コンテンツ名に制御情報を含めた CCN アプリケーションに関する研究、および分断ネットワークにおける情報通信基盤に関する研究について述べる。

2.1 コンテンツ名に制御情報を含めた CCN アプリケーションに関する研究

CCN では名前を経路制御識別子としており、その定義空間の広さや意味付与の簡便さといった観点から、コンテンツ名をルーティングだけでなく制御情報としても利用する研究が多くなされている。文献 [1] に挙げられている VoCCN は VoIP を CCN 上に実現したものであり、VoIP における SIP を用いた音声リンクの確立を図 2 のようにコンテンツ名で表現することで可能としている。これにより VoIP で必要であった制御用の経路と音声データ用の経路を一本化することができ、通話するという目的をシンプルに達成することができる。文献 [2] に挙げられている Dash over CCN はストリーミング配信に CCN を適用したものであり、図 3 のようにインタレストパケットをセグメント毎に取得することでセグメント単位で動画・音声のビットレートや解像度等を変更することが可能となる。これにより、ネットワークの輻輳状態によってコンテンツの品質を柔軟に変更することができる等、利用効率の向上が期待される。また文献 [3] で挙げられている CCN Homenet では図 4 のように FIB を拡張してコンテンツ名のプレフィックスに対する Action-flag を設定することで、コンテンツ（サービス）毎にアクセス制御やファイアウォール等の様々なポリシーを設定できる。

このように CCN は IP では困難なネットワーク層における柔軟な制御を可能とするが、既存の研究の多くはエンドノードおよび中継ノードにおけるデータ処理に関する制御が主として考えられているのみであり、機器の物理移動を伴う制御についてはあまり検討されていない。本報告で提案する自律移動可能なルータは CCN における新たな制御の可能性を示すとともに、分断ネットワークにおいて CCN の利点を活かした効率的な情報流通基盤を担うと考えられる。

2.2 分断ネットワークにおける情報流通基盤に関する研究

分断ネットワークにおける情報流通基盤として、DTN に関する研究が現在盛んに行われている。文献 [7] では資源制約の強いデバイス上で DTN のルーティングを行うためのプロトコル設計およびオープンビーコンタグを用いた実装実験が行われており、文献 [8] では災害時に分断された独立ネットワークを UAV がランダムに巡回することで検知し、DTN プロトコルを用いて独立ネットワーク間でのデータコピーを行う研究がなされている。また、近年では DTN に情報指向ネットワークのパラダイムを導入した研究も盛んに行われている。

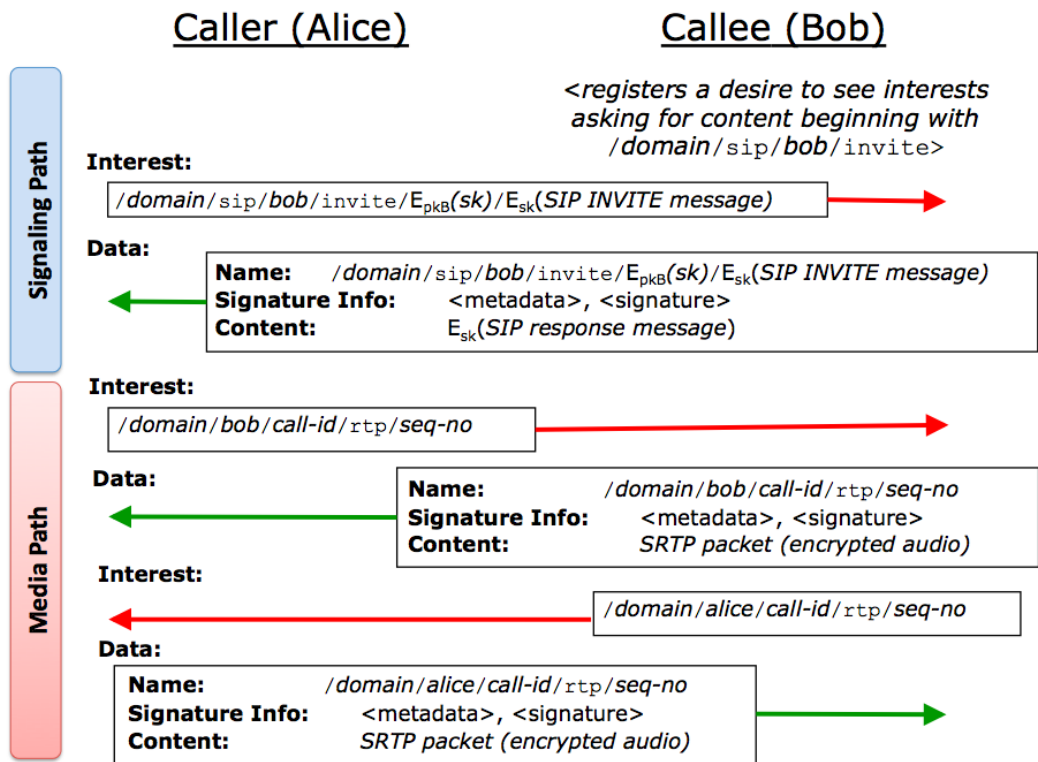


図 2: VoCCN における通信プロトコル (文献 [1] の Figure 4 から引用)

文献 [9] では情報指向ネットワークと DTN の原理を統合するための議論がなされており，文献 [10] ではさらに CCN の苦手とするプッシュ型の通信もサポートされている。

ところが DTN はあるノードと通信できるようになれば通信する，といった受動的な通信プロトコルであるため，本質的に経路制御するものでなく，経路やコンテンツ転送において最適化を図ることができない．一方で本報告で提案するルータの実移動を伴う情報指向ネットワークではルータの能動的な移動制御が可能であり，さらに情報指向ネットワークの利点であるキャッシングやコンテンツ名の活用等によって，経路制御やコンテンツ転送の最適化が可能となる．これにより分断ネットワークにおけるメッセージ到達率・到達時間といった

The name for a DASH segment

http://www-itec.uni-klu.ac.at/BigBuckBunny/bunny_2s/bunny_2s_150kbit/bunny_2s1.m4s

The name for a DASC segment

ccnx:/DASC/http://www-itec.uni-klu.ac.at/BigBuckBunny/bunny_2s/bunny_2s_150kbit/bunny_2s1.m4s

図 3: Dash over CCN におけるセグメント毎のコンテンツ名 (文献 [2] の Figure 3 から引用)

Service	Action-flag	Next hop
/entertainment/media-center/video/service/	0x11 → Action: {Access Control list {x,y, z}}	F1, F2
/climate/thermostat-1/service/	0x11 → Action: {Access Control list {x,y,z}}	F1

図 4: CCN Homenet におけるポリシーベースルーティングに拡張された FIB (文献 [3] の Fig. 4 から引用)

通信性能の向上はもちろん, 災害時等において高いメッセージ普及率を達成することも可能となる.

3 ルータの実移動を伴う情報指向ネットワークの提案

3.1 提案ネットワークの概要

本報告で想定する環境は、通信インフラの未整備や災害等によって相互接続性をもたない、あるいは相互接続性をもたなくなったネットワークが図5のように物理的に分散して独立している状況である。各独立ネットワークは主として無線LANで構成されており、その中心となるゲートウェイ（GW）とそれに接続されているノード、あるいは図5中のNetwork 0のようにホストのみによって構成されている。すなわち、想定する分断ネットワークはゲートウェイを中心として構成される独立ネットワーク（Intra-region Network）とその独立ネットワークを相互接続するための地点間ネットワーク（Inter-region Network）の2階層により構成されるものとし、本報告ではこの地点間ネットワークをCCNルータとUAVを組み合わせた飛行ルータ（AR）により実現する。飛行ルータの動作は、地点間ネットワークにおけるルーティングに必要な経路情報を収集する巡回モードと、実際に地点間ネットワークにおいてコンテンツをルーティングする配送モードから構成される。図6に飛行ルータにおける巡回モードと配送モードの動作概要を示す。巡回モードでは地点間ネットワークを巡回することで各独立ネットワークのゲートウェイにて収集されている経路情報を収集し、地点間ネットワークにおける経路情報の生成を行う。そして配送モードでは巡回モードで生成した経路情報を参照することにより、実際に地点間ネットワークにてコンテンツのルーティングを行う。

3.2 提案ネットワークにおける経路情報

本節では独立ネットワークおよび地点間ネットワークのルーティングに必要な経路情報を定義し、その生成方法について言及する。

3.2.1 経路情報の定義

本項では独立ネットワークと地点間ネットワークにおいてシームレスなネームベースルーティングを行うための経路情報について定義する。ここで独立ネットワークにおけるルーティングは一般的なCCNネットワークと同様の方法で実現できるため、飛行ルータの物理移動を伴う地点間ネットワークのルーティングのみを議論の対象とする。本報告ではこの地点間ネットワークのルーティングの実現のために、飛行ルータにおいてFIBおよびPITの次ホップに独自の移動制御用インターフェイスを用いることを提案する。これにより、あるコンテンツ名を含むインタレストパケットおよびデータパケットの宛先ノードが自身のネッ

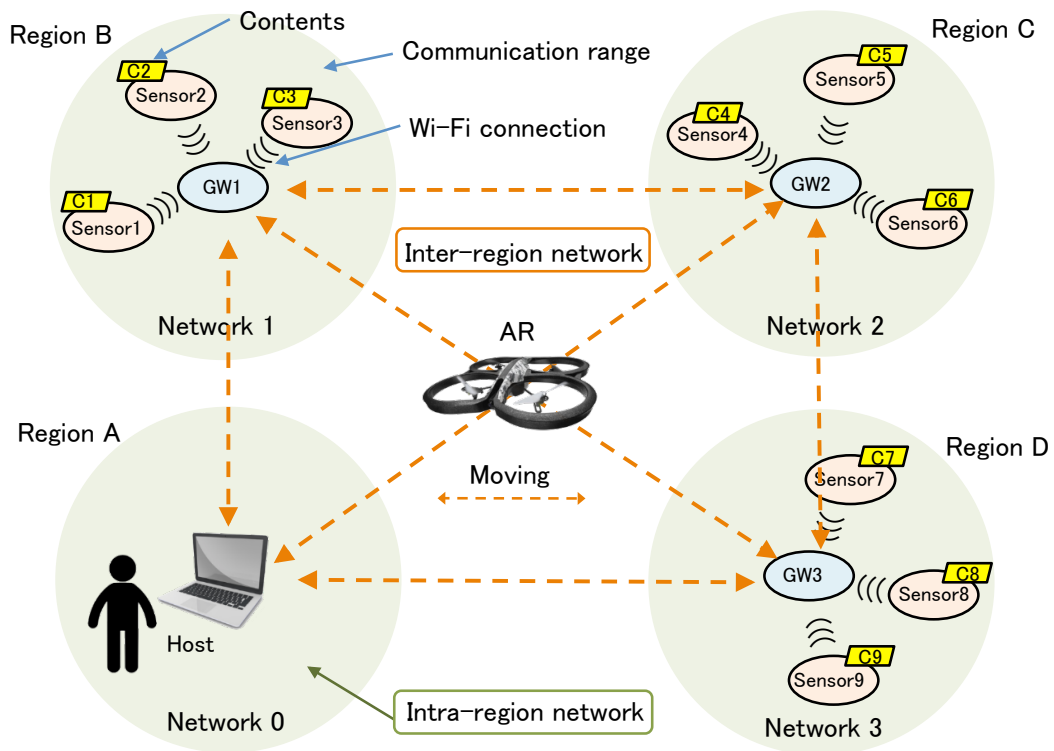


図 5: 提案ネットワークが想定する環境

トワークに存在しない時、宛先ノードの位置まで物理的に移動するといった処理を仲介させることが可能となる。具体的な移動処理については、提案する移動制御用インターフェイスの ID に対しその移動先として各独立ネットワークを構成するゲートウェイの緯度経度を対応付けたインターフェイステーブルを用いることで行う。

このように、既存の CCN ルーティングを拡張する形によってルータの移動制御を行うことで、ユーザはコンテンツの提供場所がネットワーク的に接続されているかどうか意識することなく、通常の CCN ネットワークと同様の手順でシームレスにコンテンツを取得することが可能となる。また、ここで述べる移動制御用インターフェイスとは飛行ルータで動的に生成される仮想的なものであり、独立ネットワークの数が増えても柔軟に対応できることや、本報告において CCN の実装に用いる CCNx [11] のアーキテクチャを変更せずに移動制御を加えられること等が利点となる。

3.2.2 経路情報の生成方法

本項では分断ネットワークにおける上記の移動制御を含めた経路情報の生成方法について示す。各独立ネットワークにおけるゲートウェイは自ネットワークに接続されているノード

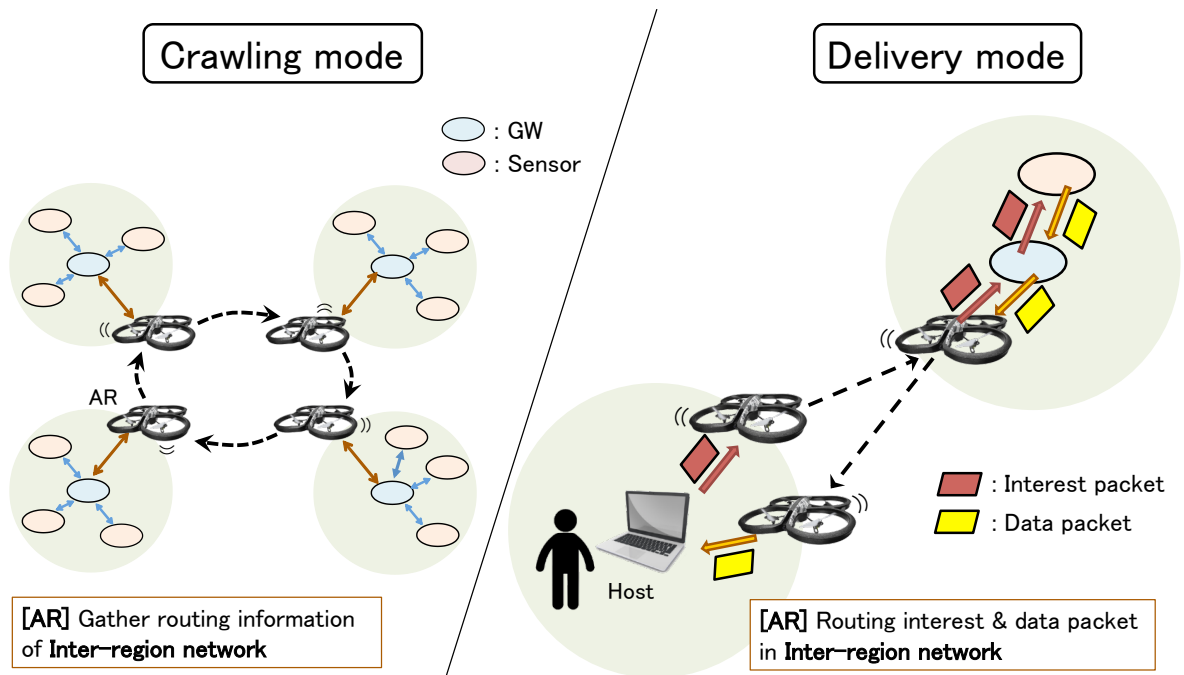


図 6: 飛行ルータにおける巡回モードと配送モードの動作概要

が保持しているコンテンツの情報を問い合わせ、その独立ネットワークにおける FIB を生成する。一方で飛行ルータは巡回モードにおいて各独立ネットワークがどのようなコンテンツを有するかを、地点間ネットワークを巡回して各ゲートウェイと経路情報の交換を行うことにより収集する。経路情報の交換は、飛行ルータが各独立ネットワークに接続された時に行われ、ゲートウェイの緯度経度とゲートウェイの収集したコンテンツリスト（すなわち FIB エントリ）を受け取り、飛行ルータは地点間ネットワークにおける FIB 及び移動制御に用いる仮想インターフェースのテーブルを更新する。例として、図 5 の分断ネットワークを対象とした巡回モードの動作を図 7 に示す。図 7 では飛行ルータが Network 0, Network 1, Network 2, Network 3 の順で各独立ネットワークに接続した場合を仮定しており、飛行ルータは各独立ネットワークに接続される度にゲートウェイの緯度経度とそのゲートウェイが所持する FIB エントリを取得し、自身の FIB とインターフェーステーブルを図のように更新していく。このとき、自身の FIB に登録するコンテンツの次ホップには緯度経度を取得したゲートウェイに対応付けられた仮想インターフェースの ID を用いる。これにより、例えばコンテンツ C8 を要求するインタレスト packets を受信した際に GW3 の緯度経度である LAT/LON3 に移動する制御を介入させることが可能となる。

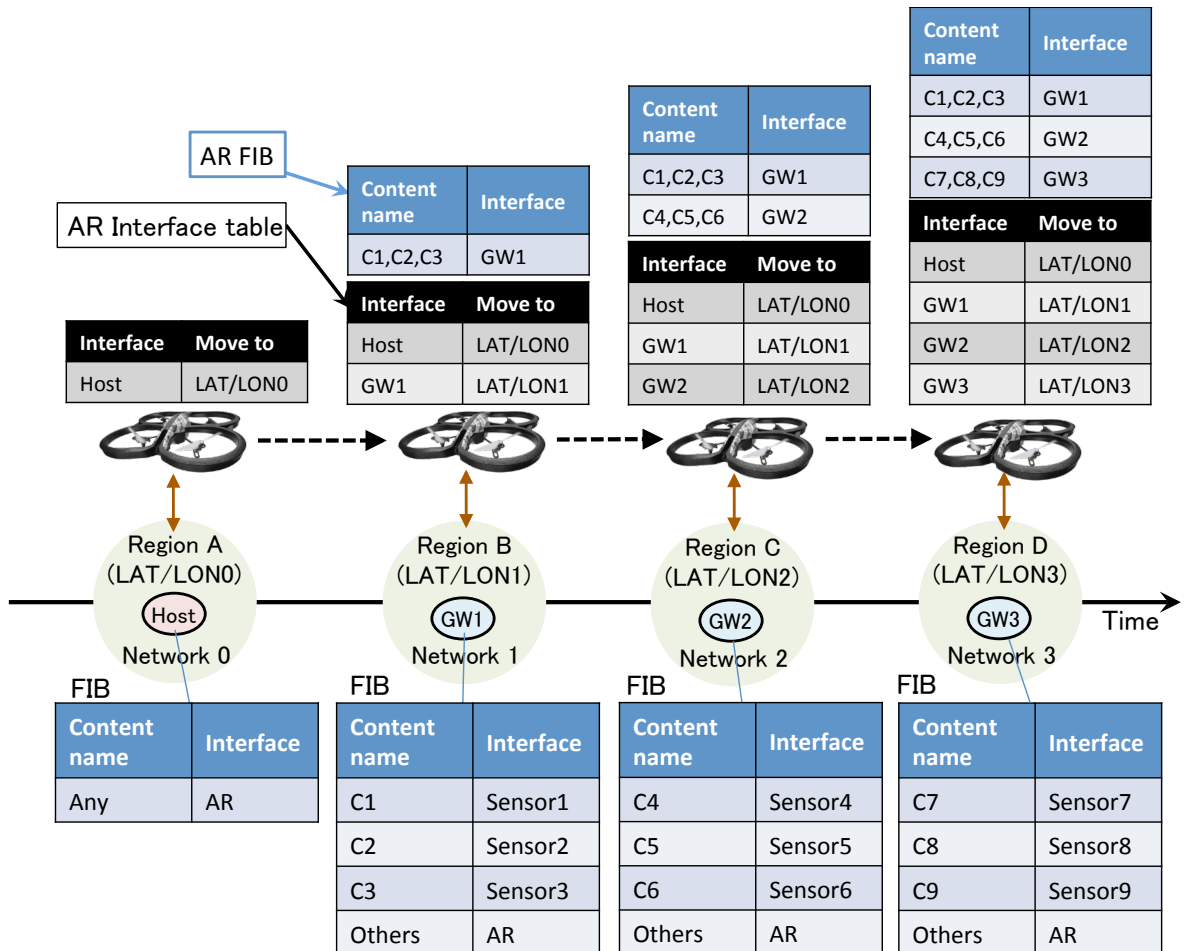


図 7: 巡回モードにおける経路情報の生成例

3.3 提案ネットワークにおけるコンテンツの取得

本節では地点間ネットワークを介したコンテンツの取得方法について議論する。各独立ネットワークのゲートウェイはインタレストパケットを受信した際に自ネットワークにそのコンテンツ（目的コンテンツ）がなければ飛行ルータにインタレストパケットを送信する。インタレストパケットを受信した飛行ルータは配送モードへと移行し、自身の FIB とインターフェイステーブルを参照して目的コンテンツのあるネットワークに移動・接続することで目的コンテンツを得る。そして飛行ルータは目的コンテンツのデータパケットを受信すると、今度は PIT とインターフェイステーブルを参照し、コンテンツ要求のあったネットワークに戻りデータパケットをコンテンツ要求ホストに返送する。ここでコンテンツとは、各ノードが有する情報全般やセンサ等の情報を含んだものである。例として図 5 において

Region A にいるホストがコンテンツ C8 を取得する手順について図 8 に示し、その詳細を以下に述べる。なお、この図では独立ネットワーク (Network 3) 内のルーティングは省略している。

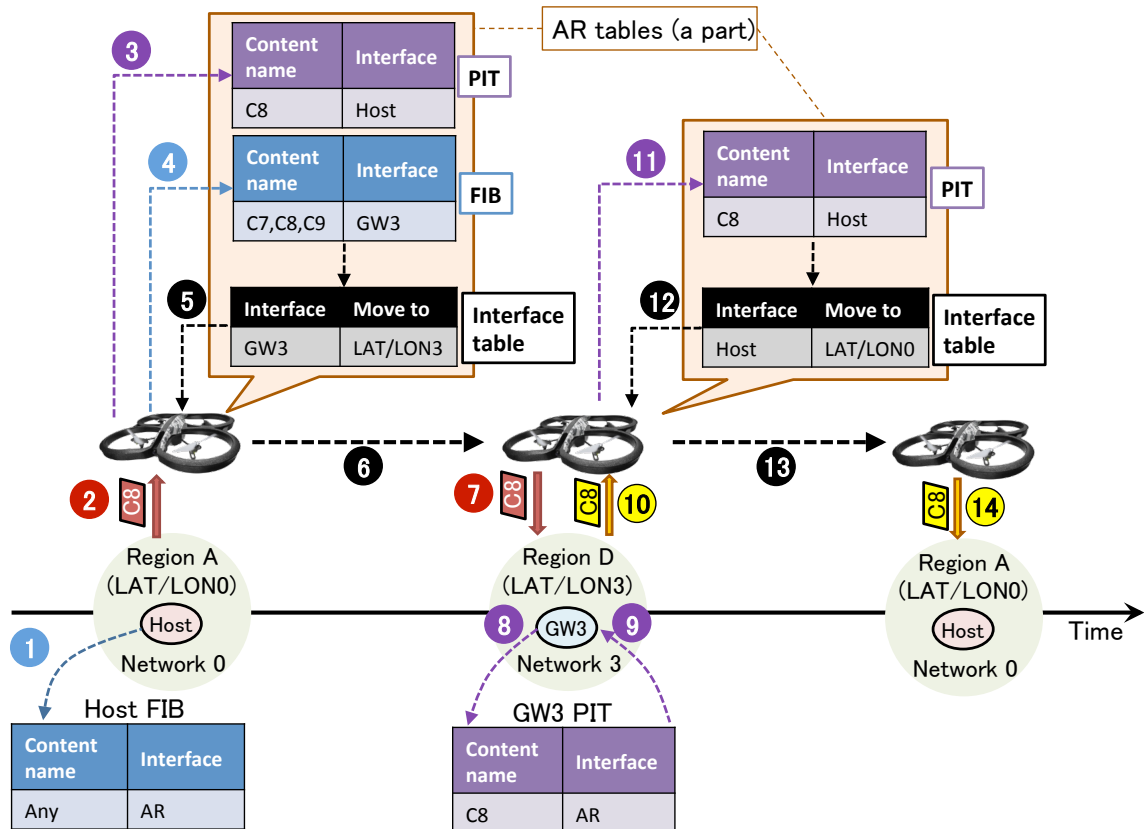


図 8: 配送モードにおけるコンテンツ取得例

- (1) Host は C8 を要求するインタレストパケットを生成し、FIB を参照することで次ホップが飛行ルータであることを得る。
- (2) Host は飛行ルータに接続可能となった段階でインタレストパケットを送信する。
- (3) インタレストパケットを受信した飛行ルータは PIT に C8 と Host の対応付けを記録する。
- (4) 飛行ルータは FIB を参照することで宛先インターフェイスが GW3 であることを得る。
- (5) 飛行ルータはインターフェイステーブルを参照することで GW3 の緯度経度が LAT/LON3 であることを得る。
- (6) 飛行ルータは LAT/LON3 に移動し、Network 3 に接続する。

- (7) 飛行ルータは Network 3 に接続すると、インタレストパケットを GW3 に転送する.
- (8) インタレストパケットを受け取った GW3 は PIT に C8 と飛行ルータの対応付けを記録する.
- (9) GW3 は PIT を参照し、(自身のネットワークから得た) C8 のデータパケットの返送先が飛行ルータであることを得る.
- (10) GW3 は飛行ルータに接続可能となった段階でデータパケットを返送する.
- (11) データパケットを受信した飛行ルータは PIT を参照することで宛先インターフェイスが Host であることを得る.
- (12) 飛行ルータはインターフェイステーブルを参照することで Host の緯度経度が LAT/LON0 であることを得る.
- (13) 飛行ルータは LAT/LON0 に移動し、Network 0 に接続する.
- (14) 飛行ルータは Network 0 に接続すると、データパケットを Host に転送する.

4 提案ネットワークの設計および実装

本章では3章で提案したネットワークアーキテクチャを実現するために、飛行ルータおよび各独立ネットワークを構成するゲートウェイの設計を行い、その実装方法について述べる。

4.1 飛行ルータの設計

UAVに小型のコンピュータを搭載し、UAVにおける移動制御およびCCNにおけるルーティングを搭載したコンピュータ上で行うことで自律移動可能なCCNルータを作成する。本節では地点間ネットワークの移動制御に用いる仮想インターフェイスとして仮想プロキシ方式を提案・設計し、さらに飛行ルータにおける巡回モードと配送モードの動作についても設計する。

4.1.1 仮想プロキシ方式の設計

飛行ルータは自己位置推定モジュールと機体移動モジュールを組み合わせることによって目的地を入力とする自律移動制御を行うが、CCNのルーティングテーブル(FIB/PIT)とこの自律移動制御を連携させる仮想インターフェイスとして本報告では仮想プロキシ方式を提案する。図9に仮想プロキシ方式の動作概要を示す。

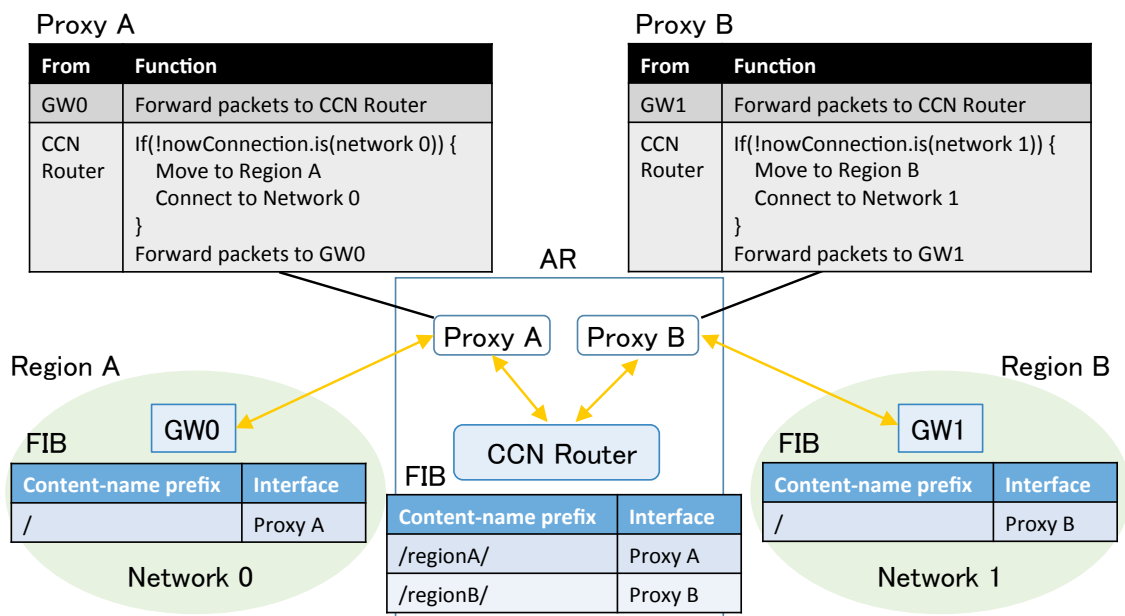


図 9: 仮想プロキシ方式の動作概要

仮想プロキシは分断ネットワーク全体で検知された独立ネットワークの数だけ飛行ルータ上に立ち上げられ、各ネットワークに対し1つのプロキシが飛行ルータとの通信を専属して仲介する。各プロキシは自身に送られてきたパケットの送信元情報のみを参照し、送信元が担当のネットワークであれば飛行ルータ内のCCNルータに、送信元が飛行ルータ内のCCNルータであるならば担当のネットワークにパケットを転送する。このとき飛行ルータが担当のネットワークに接続されていない場合は自律移動制御モジュールを起動させ、担当ネットワークに接続できる地点（ネットワークを構成しているゲートウェイの緯度経度）まで移動してからパケットの転送を行う。

4.1.2 巡回モードにおける動作設計

巡回モードでは飛行ルータは分断ネットワーク全体をランダムに飛行し、独立ネットワークのゲートウェイが発する無線電波を検知するとそのゲートウェイとの接続を確立する。そして接続が確立され次第ゲートウェイの緯度経度を取得し、その独立ネットワークと飛行ルータにおけるCCN機構の通信を仲介する新たな仮想プロキシを設定する。またこのとき、ゲートウェイからその独立ネットワーク内にあるコンテンツリストを取得し、自身のFIBエントリとしてそれらの登録も行う（登録したコンテンツ名に対する次ホップは新たに設定された仮想プロキシとする）。なお検知した独立ネットワークに対する仮想プロキシが既に確立されていた場合においても、独立ネットワーク内の経路情報が更新されていれば自身のFIBも更新する。

4.1.3 配送モードにおける動作設計

配送モードでは分断ネットワークにおけるコンテンツの取得を行うが、通常のネットワーク利用では宛先が同一であるパケットが連続して送信されるケースが考えられる。このため、配送モードでは仮想プロキシにてパケットのバッファリング機能を提供する。バッファリング機能はあるパケットの宛先ネットワークが飛行ルータと接続されていないとき、各仮想プロキシにパケットが N 個バッファされてからそれぞれ自律移動制御を開始するものである。また、バッファリング機能によってパケットの送受信が頻繁に行われないネットワークにて飛行ルータが長時間待機することがないように仮想プロキシにおいて担当ネットワークに対する最大接続時間を設ける。この最大接続時間は、仮想プロキシが担当ネットワーク宛での最初のパケットを受信してから T_{IN} 秒経過していれば自律移動制御を開始するものである。これらのバッファリング機能と最大接続時間の設定により地点間ネットワークにおける飛行ルータの移動回数の低減ならびにコンテンツ取得時間の短縮が期待される。また、各

プロキシによる移動処理が同時期に重なった場合には移動処理のスケジューリングにより排他制御を行う。

4.2 独立ネットワークにおけるゲートウェイの設計

各独立ネットワークを構成するゲートウェイは予め自ネットワークにおける経路情報を生成し、ゲートウェイは飛行ルータと接続可能となった時点で収集したコンテンツリストを飛行ルータに通知する。ここで、ゲートウェイは飛行ルータと接続可能となるまで地点間ネットワーク宛てのコンテンツを保持しておく必要があるが、本報告ではゲートウェイにおいても飛行ルータと同様に仮想プロキシ（便宜上、GW Proxy と呼ぶ）を設定することでこれに対応する。GW Proxy はゲートウェイ毎に一つ設定され、ゲートウェイ内の CCN ルータと飛行ルータの仮想プロキシ（便宜上、AR Proxy と呼ぶ）の通信を仲介する。図 10 に GW Proxy の動作概要を示す。GW Proxy は AR Proxy と同様にパケットの送信元情報のみを参照し、AR Proxy から受信したパケットはゲートウェイ内の CCN ルータに、ゲートウェイ内の CCN ルータから受信したパケットは AR Proxy に送信する。ただし、AR Proxy に送信する際には飛行ルータとの接続が確立されるまで待機する（接続確認を T_{IRN} 秒間隔で行う）。また飛行ルータとの接続待機中に受信したパケットについては GW Proxy にてバッファリングしておき、飛行ルータと接続された時点でバッファされたパケットを全て送信する。

このようにゲートウェイにおいてもプロキシを採用することで、飛行ルータの場合と同様に既存の CCN 通信基盤を変更することなく、ゲートウェイにおける地点間ネットワーク宛てパケットの損失防止や飛行ルータにおけるバッファリング待機時間の短縮が可能となる。なお、GW Proxy を設定する際にはゲートウェイ内の CCN ルータの FIB において、自ネットワークに存在しないコンテンツに対する次ホップを AR Proxy でなく GW Proxy にする必要がある。

4.3 提案ネットワークの実装

本節では提案したネットワークの実装を行う。なお本報告では基礎的な検証実験を想定しており、分断ネットワークにおいてコンテンツを取得できることを簡便に確認するため、飛行ルータにおける仮想プロキシ（4.1.1 項）および配送モード（4.1.3 項）のみを実装の対象とする。

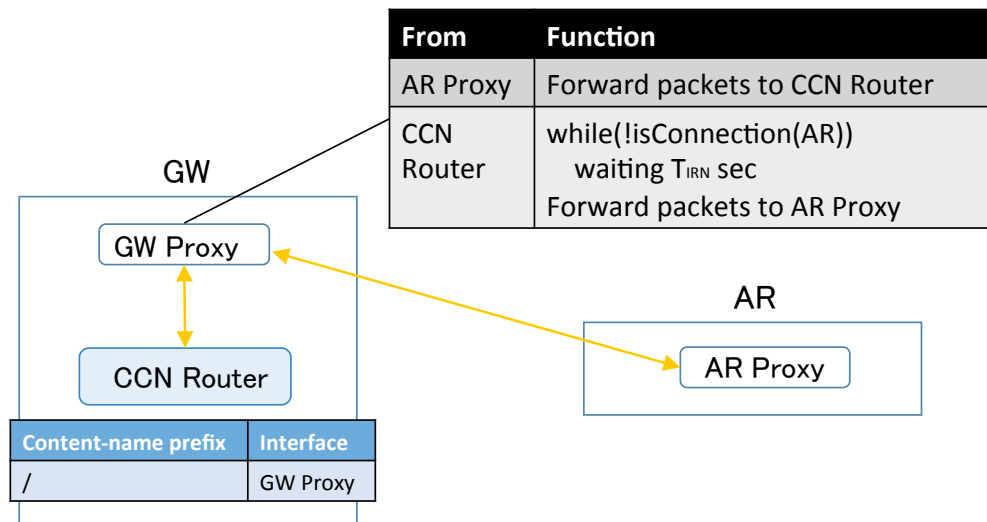


図 10: GW Proxy の動作概要

4.3.1 飛行ルータの実装構成

飛行ルータに用いる UAV には Parrot 社製の AR.Drone 2.0 Power Edition [12] を使用する。AR.Drone は Wi-Fi 経由で送られたコマンドにより空中飛行制御可能な小型のクアッドコプターである。Wi-Fi 通信機構として AP (Access Point) が搭載されているため、クライアントは Wi-Fi 接続により AR.Drone の制御を行うことができる。さらに本報告では地点間ネットワークの構成、すなわち飛行ルータと独立ネットワークにおけるゲートウェイの接続にもこの AP を用いる。また、AR.Drone 上に搭載する小型のコンピュータとしては Raspberry Pi Model B+ [13] を採用し、CCN の通信および AR.Drone の移動制御を行う。Raspberry Pi は英国のラズベリーパイ財団により開発された ARM プロセッサ型のシングルボードコンピュータであり、小型・軽量に加え USB 給電で動作する省電力性を併せ持つ。使用した AR.Drone 2.0 Power Edition および Raspberry Pi Model B+ の仕様をそれぞれ表 1, 表 2 に示す。また、CCN の通信の実現には PARC によって開発された CCN プロトコルのリファレンス実装である CCNx [11] を用いる。CCNx では FIB/PIT における宛先インターフェイスとして IP アドレスとポート番号の組を使用しているため、このポート番号を飛行ルータにおける仮想プロキシの ID として用いることで提案手法を実装する。

飛行ルータの具体的な構成図を図 11, 実際の外観を図 12 に示し、使用するハードウェアおよびソフトウェアについて表 3, 表 4 にまとめる。Raspberry Pi については AR.Drone の USB 端子から電源を供給し、USB 型無線 LAN ドングルを用いて AR.Drone の AP と接続することにより、CCNx に関する通信、AR.Drone に対する制御信号の送信、そして制御に必要な AR.Drone のステータス情報の受信を行う。また、AR.Drone の自己位置推定には

表 1: AR.Drone 2.0 Power Edition の仕様

CPU	32bit ARM Cortex A8 1GHz
RAM	1GB
OS	Linux 2.6.32 (Busybox)
無線 LAN	IEEE 802.11b/g/n
寸法	512x451mm
重量	380g
バッテリー	11.1V, 1500mAh
その他	USB port x 1

表 2: Raspberry Pi Model B+ の仕様

SoC	Broadcom B BCM2835 (CPU, GPU, DSP, SDRAM 内蔵)
CPU	700 MHz ARM1176JZF-S
RAM	512MB
OS	Raspbian
寸法	85x56mm
重量	45g
電源	5V 600mA MicroUSB
その他	MicroSD card slot x 1, USB 2.0 port x 4

Raspberry Pi に装着した USB-GPS ドングルから得た緯度経度を使用し、開発言語には全て Java を用いる。

4.3.2 仮想プロキシの実装

本項における仮想プロキシとは飛行ルータにて実装されるものを指す。仮想プロキシは担当の独立ネットワークにおけるゲートウェイと飛行ルータの CCNx デーモン（仮想的な CCN ルータ. ccnd と呼ばれる）の通信を中継し、ccnd からパケットを受信した場合は担当の独立ネットワークに移動・接続し、受信したパケットを転送する。ここで配送モードにおける効率的なルーティングのために担当ネットワークに接続されていない状態であれば ccnd から受信したパケットをある一定数 N までバッファリングする（担当ネットワークに接続されている場合はバッファリングせずにそのまま転送する）。仮想プロキシのクラス（Proxy クラス）とその関連クラスのクラス図（一部）を図 13 に示す。

Proxy クラスでは仮想プロキシの処理、ProxyConfig クラスは仮想プロキシの設定項目、

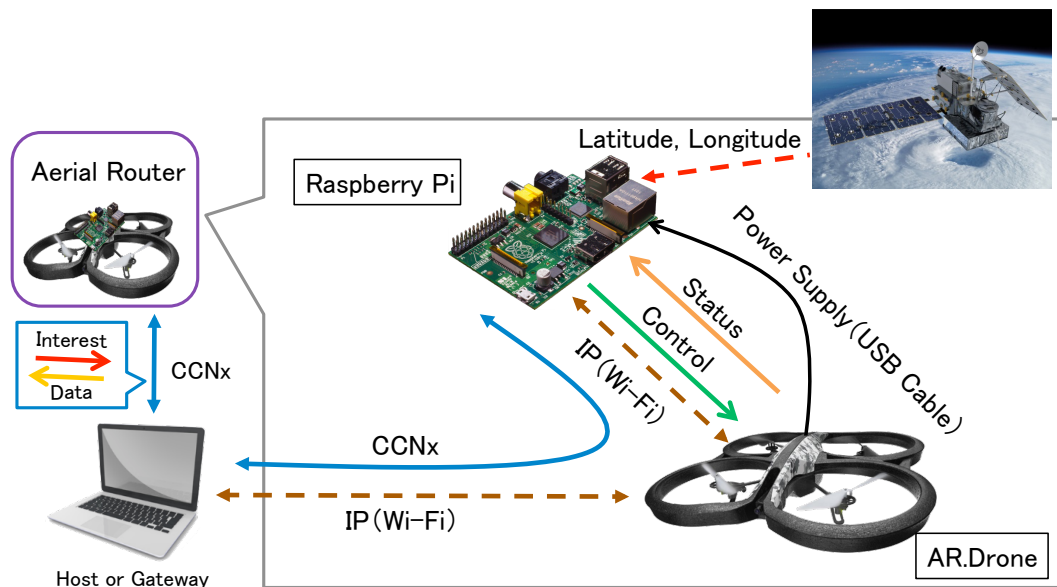


図 11: 飛行ルータの構成図

Move クラスは AR.Drone の自律移動制御， ARDrone クラスは AR.Drone におけるステータス情報や制御処理を定義したものである（全て自作のクラスであるが， ARDrone クラスにおいてはその実装の大部分が ARDroneForP5 [14] によるものである）。仮想プロキシは ProxyConfig クラスのフィールドを元に設定を行う。 ProxyConfig クラスの設定項目としては ccnd および地上ゲートウェイからパケットを受信するポート番号（from_ccnd_listenport, from_gw_listenport）， それらにパケットを送信するための宛先 IP アドレスと宛先ポート番号（to_ccnd_ipaddress, to_ccnd_port, to_gw_ipaddress, to_gw_port）， そして担当ネットワークを構成するゲートウェイの緯度経度（gw_gps_location）がある。なお， トランスポート層プロトコルには UDP を使用し， 断続的に通信できるようコネクションレスの通信を行う。仮想プロキシ（Proxy クラス）は relay_packet() メソッドにおいて ccnd から（すなわち from_ccnd_listenport にて）パケットを受信すると Move クラスにおける now_network を参照し， 自身の担当するネットワークが現在接続されているネットワークであれば担当ネットワークのゲートウェイにパケットを中継， そうでなければ buffer_list に受信したパケットを格納する。そして格納したパケットの数が N に達するか， 1 つ目のパケットを受信してから T_{IN} 秒経過した場合に仮想プロキシは proxy_config を引数として Move クラスの moveTo() メソッドを呼び出し， 移動制御を依頼する。 Move クラスは移動処理を終えると isNetworkConnect() メソッドを用いて担当ネットワークに接続したかどうかを確認し， 接続されるまで待機する。担当ネットワークへの接続が完了すると処理は Proxy クラスに戻り， 仮想プロキシはバッファされたパケットを担当ネットワークに転送する。



図 12: 飛行ルータの外観

4.3.3 自律移動制御の実装

本項では飛行ルータにおける自律移動制御の実装方法について述べる。自律移動制御は図 13 の Move クラスにて実装され、複数の仮想プロキシによる移動委託が同時に重なった場合は FCFS (First-Come First-Serve) に基いてスケジューリングを行う（移動を待機するプロキシは `waiting_proxylist` に格納される）。飛行ルータは自律移動制御に必要な AR.Drone の現在位置と現在の方角を次の手法で取得する。

現在位置

RXTX for Java（シリアル通信を提供する Java ライブラリ）を用いることで USB 接続された GPS ドングルから GPS データ（NMEA フォーマット）を 1 秒毎に受信し、これをパースすることで現在位置を更新する（ARDrone クラスにおける `now_gps_location`）。

現在の方角

ARDroneForP5（AR.Drone の Processing 用 API）を用いることで AR.Drone のステータス情報（`navdata`）を取得し、AR.Drone の地磁気センサにて計測される現在の方角を数百ミリ秒単位で更新する（ARDrone クラスにおける `now_drone_direction`）。

表 3: 飛行ルータのハードウェア仕様

AR.Drone 2.0 Power Edition	無線制御で空中飛行可能なクアドコプター. 無線 AP としても動作. UAV として使用.
Raspberry Pi B+	Linux が動作するシングルボードコンピュータ. CCN ルータおよび AR.Drone の移動制御に使用.
GW-USNANO2A	USB 型無線子機 (802.11n/g/b). Raspberry Pi に装着. AR.Drone との通信に使用.
GT-740FL	USB 型 GPS ドングル. Raspberry Pi に装着. 飛行ルータの自己位置推定に使用.
TS32GUSDC10	microSDHC 32GB Class10. Raspberry Pi のストレージに使用.
汎用 Micro USB ケーブル	AR.Drone から Raspberry Pi への電源供給に使用.

表 4: 飛行ルータのソフトウェア仕様

Raspbian	Raspberry Pi 用に開発された Debian ベースの Linux OS. 2014-12-24-wheezy-raspbian を使用.
Java	飛行ルータの開発言語. JDK 8u31 を使用.
CCNx for Java	CCNx の Java 実装. Version 0.8.2 を使用.
ARDroneForP5 [14]	AR.Drone の Processing 用制御 API. Java 用に変換・リビルドして使用.
RXTX for Java [15]	Java でシリアル通信を行うライブラリ. USB から GPS データを読み込む際に使用. Version 2.1.7 を使用.

図 14 は飛行ルータの自律移動制御におけるフローチャート (概要) である. 以下にその詳細を示す.

1. 経度を x , 緯度を y とする直行平面において, AR.Drone の現在位置と目的地からその差ベクトルを算出し, 北ベクトル (y 方向の単位ベクトル) と内積をとることで目的地の方角を得る (緯度経度は地球楕円体として GRS 80 を採用し, メートル単位に近似変換する).
2. AR.Drone の向いている方角が目的地の方角に対し一定の誤差 (`allowable_direction_error`) 以下であれば 5. へ遷移する. 左寄りに向いていれば 3. へ, 右寄りに向いていれば 4. へ遷移する.

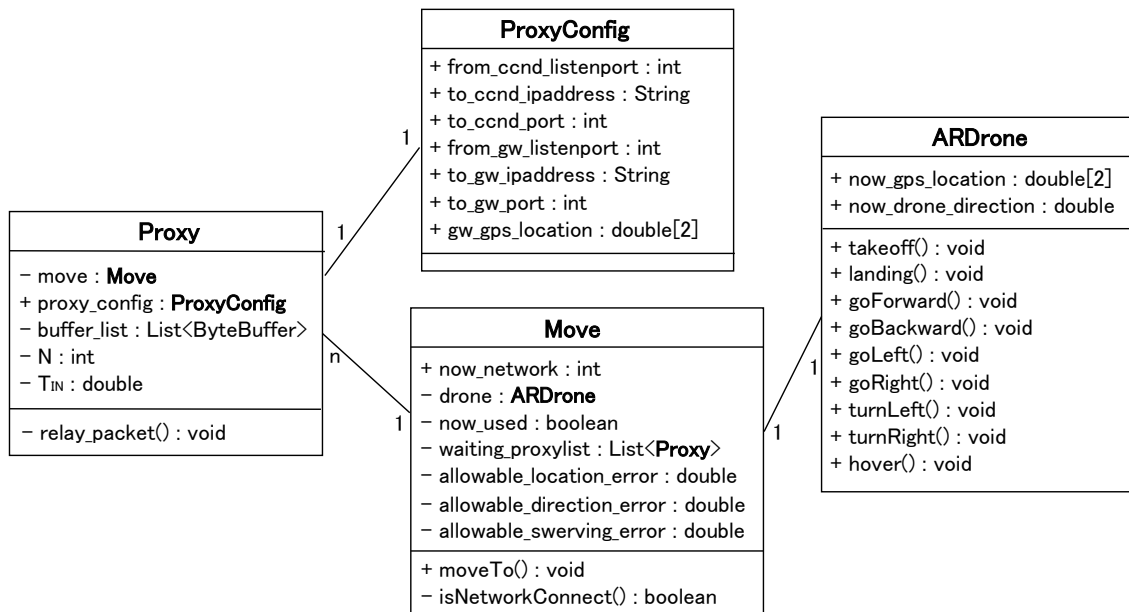


図 13: Proxy クラスとその関連クラスのクラス図 (一部)

3. AR.Drone に右回転を行う制御信号を送信する (AR.Drone が右回転を行っている最中であれば何もしない)。
4. AR.Drone に左回転を行う制御信号を送信する (AR.Drone が左回転を行っている最中であれば何もしない)。
5. AR.Drone に直進を行う制御信号を送信する (AR.Drone が直進を行っている最中であれば何もしない)。
6. AR.Drone の現在地が目的地に対し一定の誤差 (allowable_location_error) 以下であれば処理を終了する。目的地から遠ざかっていれば 1. へ遷移する。目的地に近づいていれば 7. へ遷移する。
7. 目的地と AR.Drone が前回 1. の処理 (方向転換) を行った地点との差ベクトルに対し, AR.Drone の現在地から垂線を下ろした時の垂線の長さが一定の誤差 (allowable_swerving_error) 以上であれば 1. へ遷移する。それ未満であれば 5. へ遷移する。

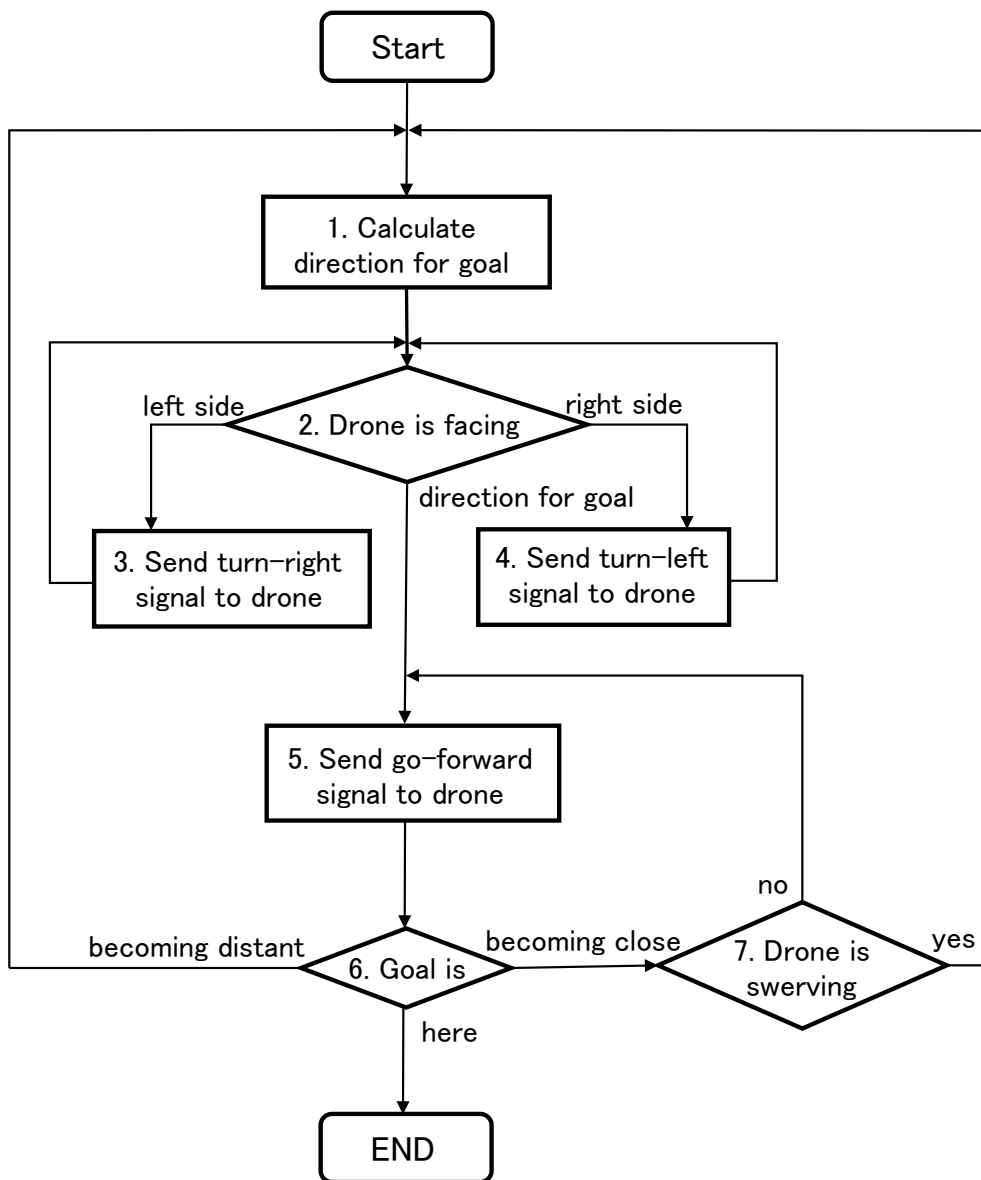


図 14: 自律移動制御におけるフローチャート (概要)

5 実験ネットワークにおける動作検証

本章では実装したネットワークの動作検証を行うため、簡便な実験ネットワークを用い地点間ネットワークのルーティングが正しく行われることを確認する。

5.1 実験環境

地点間ネットワークにおけるルーティングが正しく行われることを確認するため、大阪大学吹田キャンパスの空き地において遠隔コンテンツの取得実験を行った。以下では具体的な実験環境について述べる。

実験では、図 15 のように直接無線通信が不可能な Region A と Region B（距離差およそ 50 m）にそれぞれ 1 台ずつノードを設置し、自律移動可能な飛行ルータを介すことで両者（Node A と Node B とする）の間で CCN による通信が可能であることを検証する。コンテンツの公開および取得に用いるアプリケーションにはそれぞれ CCNx で提供されている `ccnputfile`, `ccngetfile` を用い、コンテンツとしては 4 kB 程度の画像ファイル (`apple.jpg`) を使用する。なお、Node A と Node B には CCNx のインストールされた Raspberry Pi B+ を用い、地点間ネットワークにおける IP アドレスとしてそれぞれ 192.168.1.4, 192.168.1.5 を付与する。ここで、Node A をコンテンツ取得側、Node B をコンテンツ配信側とし、飛行ルータは初期状態として Node A に接続されているとする。また、各ノードにおける FIB は予め表 5, 表 6, 表 7 のように設定しておく。表中の `interface` として指定されている (x,y) は x が宛先 IP アドレス, y が宛先ポート番号 (UDP) を指す。またポート番号 16001, 17001 はそれぞれ Proxy A の (飛行ルータの) `ccnd` に対するリッスンポート, Node A に対するリッスンポートであり, ポート番号 16002, 17002 はそれぞれ Proxy B の (飛行ルータの) `ccnd` に対するリッスンポート, Node B に対するリッスンポートである。

表 5: Node A の FIB

content-name prefix	interface
ccnx:/	Proxy A (192.168.1.3, 17001)

表 6: Node B の FIB

content-name prefix	interface
ccnx:/	Proxy B (192.168.1.3, 17002)

表 7: 飛行ルータの FIB

content-name prefix	interface
ccnx:/regionA	Proxy A (127.0.0.1, 16001)
ccnx:/regionB	Proxy B (127.0.0.1, 16002)

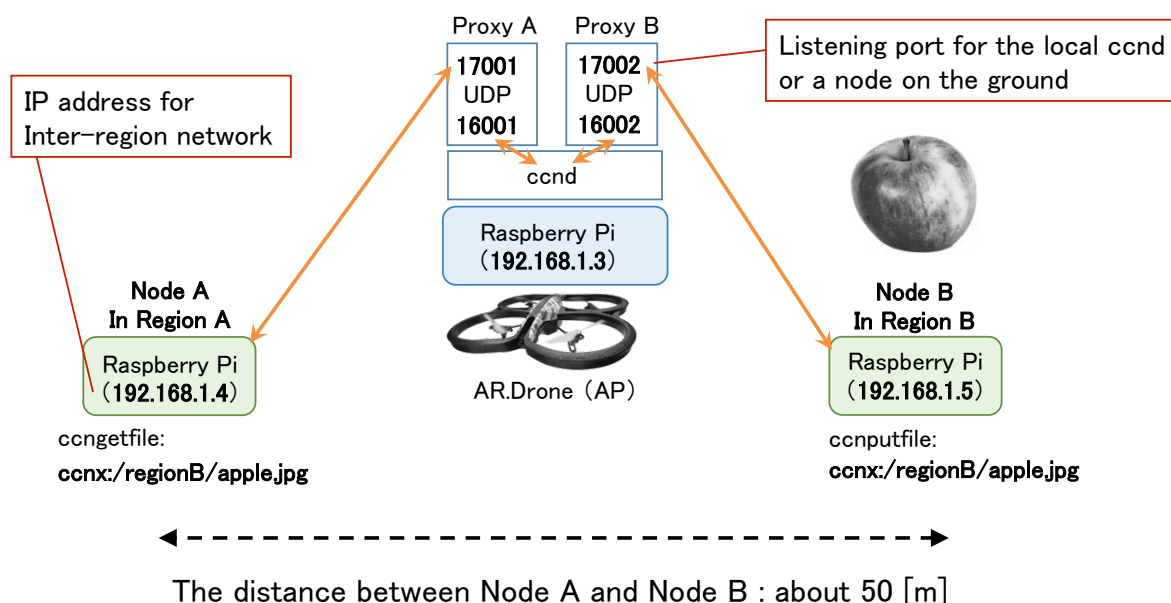


図 15: 検証実験に用いる地点間ネットワーク

5.2 実験方法

最初に実験の準備として、Node B において ccnputfile を実行し、apple.jpg を「ccnx:/regionB/apple.jpg」という名前で Node B のリポジトリに登録することでネットワークに公開する。その後 Node A において Node B (192.168.1.5) 宛てに ping コマンドを実行して両者が接続されていないことを確認し、ccngetfile によってコンテンツ名「ccnx:/regionB/apple.jpg」を要求するインタレストパケットを生成する。図 16, 図 17 にコンテンツルーティングの流れを示す。

図 16 では、ccngetfile にて生成されたインタレストパケットは各 FIB と飛行ルータ上の仮想プロキシによって Node A から Proxy A, Proxy A から飛行ルータの ccnd, 飛行ルータの ccnd から Proxy B へとルーティングされる。ここで飛行ルータはすぐに Region B へと移動せずに、Proxy B に Node B 宛ての packets が N 個バッファされるか、最初の packet を受信してから T_{in} 秒経過してから移動を開始する。飛行ルータが Region B に移動して Node B との接続を確立すると、バッファされた packets を全て Node B に転送する。そして図 17

では、Proxy A に N 個の packets がバッファされるか、最初の packet を受信してから T_{in} 秒経過してから飛行ルータは移動を開始する。飛行ルータが Region A に移動して Node A との接続を確立すると、先と同様にバッファされた packets を全て Node A に転送する。このとき Node A が受信した packets の中には目的のコンテンツ (ccnx:/regionB/apple.jpg) が含まれており、コンテンツの取得が完了する。検証実験では Region A にてノート PC から AR.Drone の AP を介して Node A の Raspberry Pi にリモートデスクトップ接続を行い、上記のルーティングを経て最終的に受信した apple.jpg が画面に表示されることを確認する。最後に、検証実験で用いる各種定数を表 8 に示す。

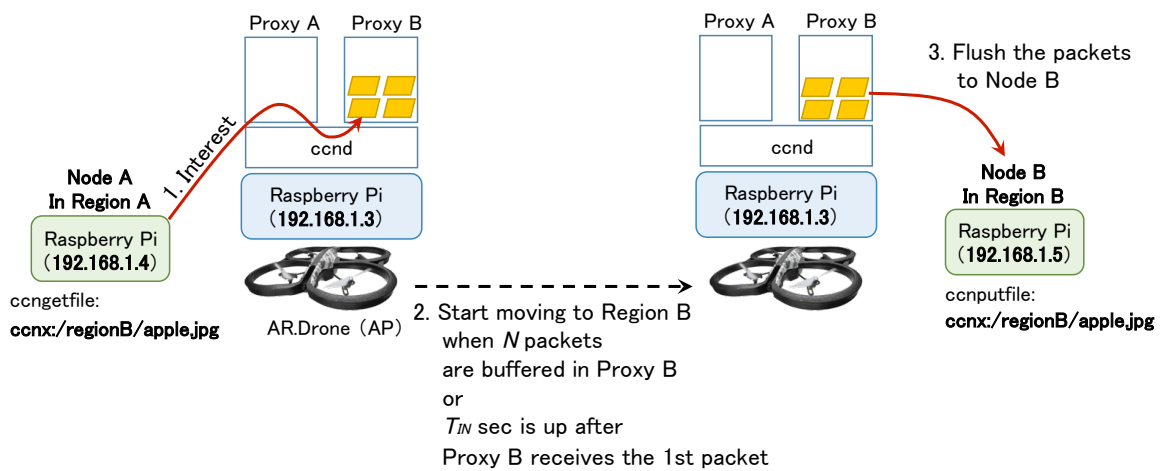


図 16: 実験ネットワークにおけるインタレストパケットのルーティング

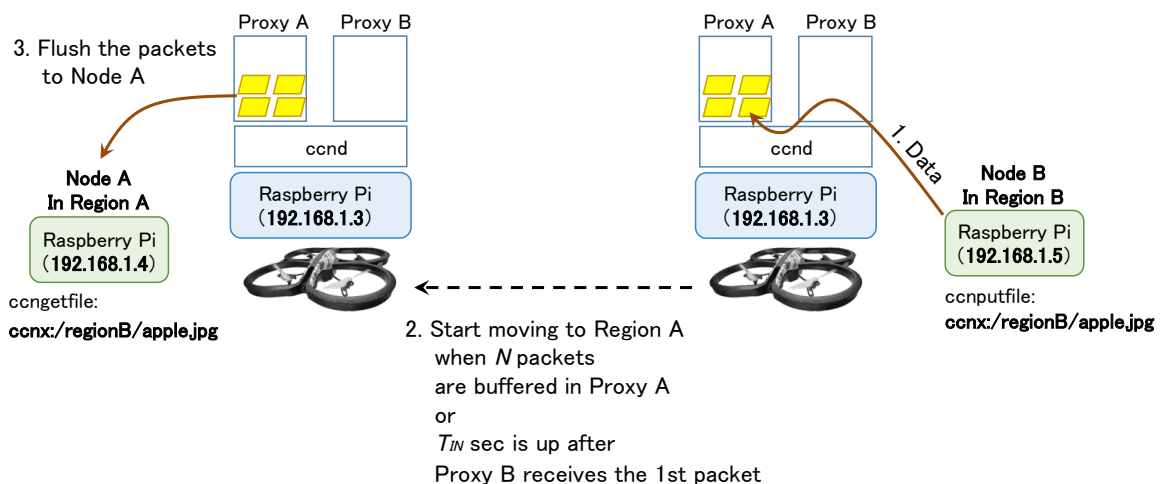


図 17: 実験ネットワークにおけるデータパケットのルーティング

表 8: 検証実験で用いる各種定数の値

定数名	値
N	10 [個]
T_{IN}	1 [s]
allowable_direction_error	2 [°]
allowable_location_error	10 [m]
allowable_swerving_error	1 [m]

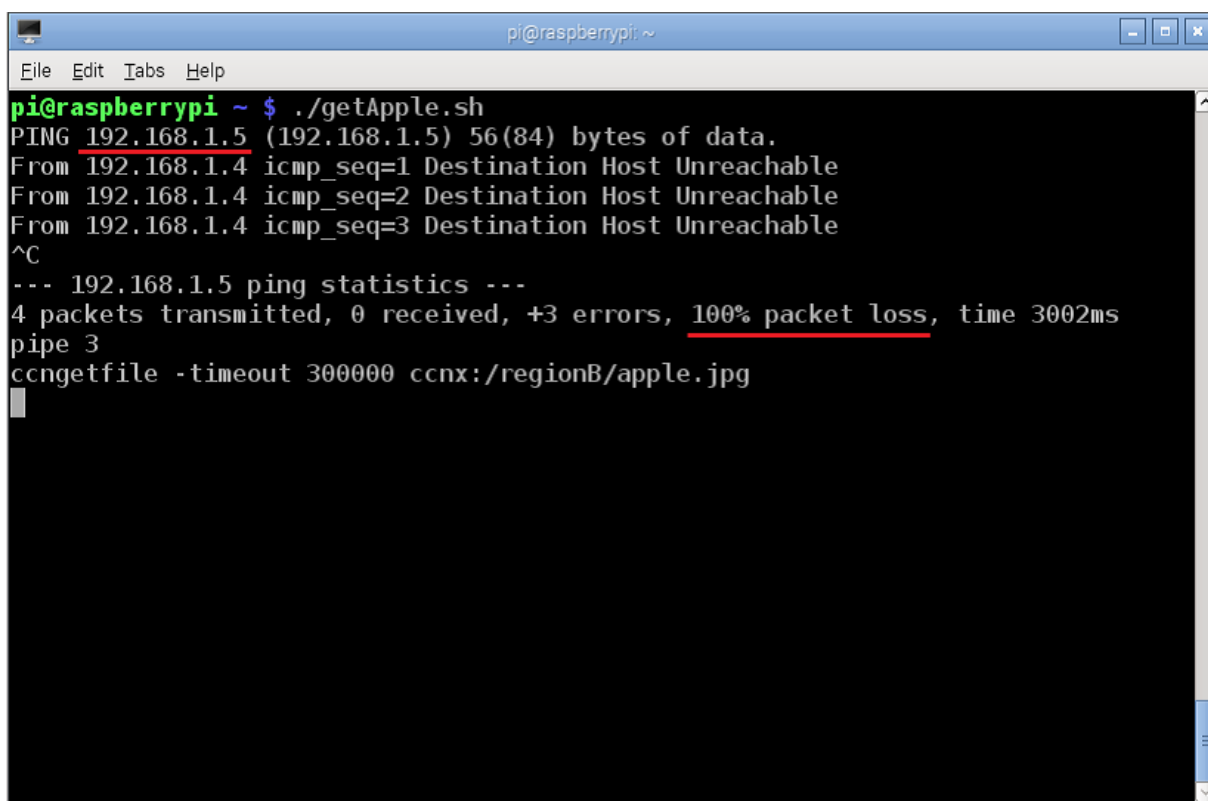
5.3 実験結果

本報告では Node A で動作させる実験用のスクリプトとして `getApple.sh` を用いる。 `getApple.sh` の処理は次の通りである。

1. Node B に向けて `ping` コマンドを実行する。
2. `ccngetfile` によりコンテンツ「`ccnx:/regionB/apple.jpg`」を取得する。なおタイムアウト時間は5分とする。
3. 画像ビューアにより画面に取得したファイル (`apple.jpg`) を表示する。

最初に Node A から Node B に `ping` コマンドを実行した結果を図 18 に示す。これにより Node A (192.168.1.4) と Node B (192.168.1.5) には接続性がないことが分かり、Node A と Node B はそれぞれ分断ネットワークにおける独立ネットワークとみなすことができる。 `ping` による接続性チェックを終えると、 `ccngetfile` の実行によって飛行ルータが離陸し、Region B まで移動した後、再び Region A に戻ってきて着陸することを確認した。このときの Node A の出力画面を図 19 に示す。図中の「Retrieved content apple.jpg got 3872 bytes.」という `ccngetfile` のメッセージは 3872 バイトの `apple.jpg` というファイルを受信したことを示しており、また `apple.jpg` が画面表示されていることから地点間ネットワークにおけるコンテンツルーティングおよび遠隔コンテンツの取得に成功していることがわかる。

これらの結果から、本報告で提案した飛行ルータを擁する分断ネットワークにおいて、ネットワーク接続環境下でないコンテンツをコンテンツ名によるルーティングによってシームレスに取得できることがわかった。また、設計した仮想プロキシ方式が CCN 通信基盤を変更せずに地点間ネットワークにおけるコンテンツルーティングを実現できることがわかった。

A terminal window titled 'pi@raspberrypi: ~' with a menu bar containing 'File', 'Edit', 'Tabs', and 'Help'. The terminal output shows a script execution: 'pi@raspberrypi ~ \$./getApple.sh'. This is followed by a ping command: 'PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.'. Three lines of ping results follow, each starting with 'From 192.168.1.4' and ending with 'icmp_seq=1', 'icmp_seq=2', and 'icmp_seq=3' respectively, all reporting 'Destination Host Unreachable'. After a '^C' interrupt, ping statistics are shown: '--- 192.168.1.5 ping statistics ---', '4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3002ms', and 'pipe 3'. The final line is 'ccngetfile -timeout 300000 ccnx:/regionB/apple.jpg'.

```
pi@raspberrypi ~ $ ./getApple.sh
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
From 192.168.1.4 icmp_seq=1 Destination Host Unreachable
From 192.168.1.4 icmp_seq=2 Destination Host Unreachable
From 192.168.1.4 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.1.5 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3002ms
pipe 3
ccngetfile -timeout 300000 ccnx:/regionB/apple.jpg
```

図 18: Node A から Node B に ping コマンドを実行した結果

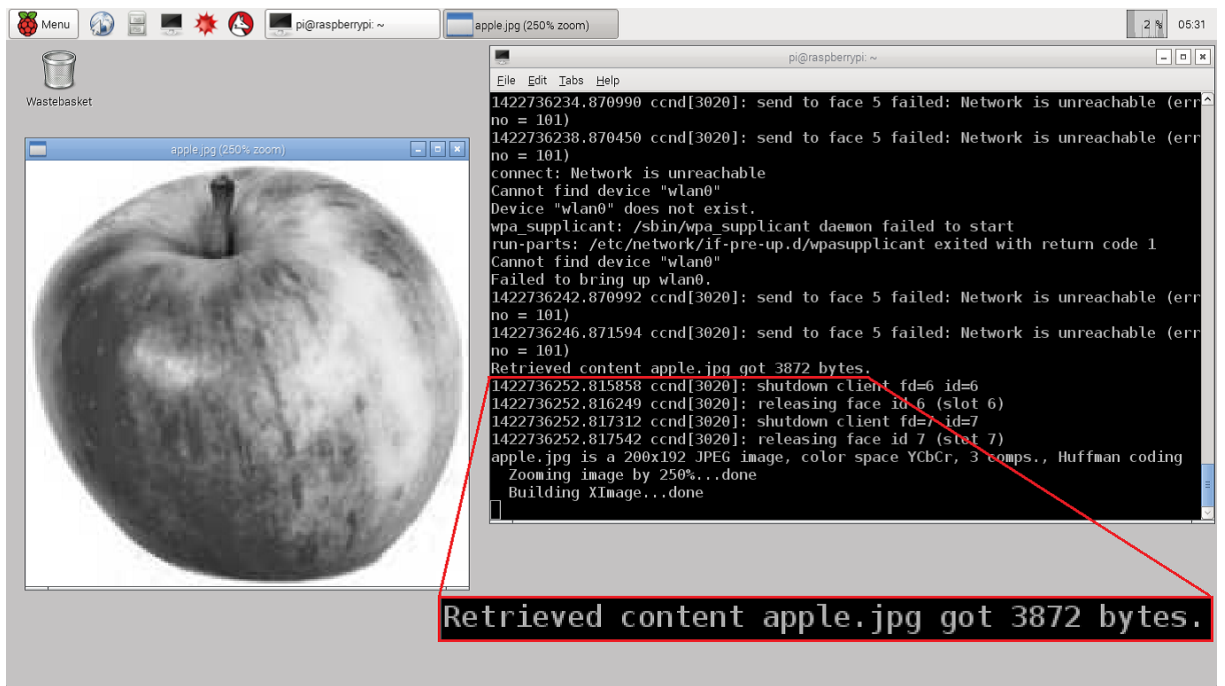


図 19: 飛行ルータが Region A に戻ってきた時の Node A における出力画面

6 おわりに

本報告では情報指向ネットワークにおけるルーティングの柔軟性に着目し、CCN ルータに通常の経路制御だけでなく物理的な移動制御を加える新たな通信基盤を提案した。ルータが物理的に移動することで、分断ネットワークにおいて広域なコンテンツルーティングが可能となり、災害あるいはインフラ未整備等によって通信できない複数地点で情報交換することが可能となる。そして提案ネットワークを実現するために、飛行ルータや独立ネットワークにおけるゲートウェイ、そしてルータの移動制御を実現させるための仮想プロキシ方式を設計・実装し、簡便な実験ネットワークを用いて設計したネットワークが実現可能であることを確認した。

今後の課題としては、複数のパケットを処理するスケジューリングやルータの移動制御を含めた経路制御の最適化、複数の飛行ルータを用いた効率的な通信手法の提案等が考えられる。また、分断ネットワークにおける既存の通信基盤に対する優位性を示すために、コンテンツの到達率や取得時間、普及率等の指標を用いて提案手法を定量的に評価することも今後の課題として挙げられる。

謝辞

本報告を終えるにあたり，ご多忙の中貴重なご指導，ご教授を賜りました大阪大学 大学院情報科学研究科の村田正幸教授，ならびに厳しくも優しいご指導で大変お力添え頂きました大阪市立大学 大学院工学研究科の阿多信吾教授に深く感謝致します。また，平素から適切なご助言やご指導を頂きました大阪大学 大学院情報科学研究科の荒川伸一准教授，大下裕一助教授，および大阪大学 大学院経済学研究科の小南大智助教授にこの場をお借りしまして心よりお礼申し上げます。また研究チームの先輩として，大岡睦氏には日頃から優しく分かりやすいご指導を頂き，北出雄麻氏には研究以外においても幅広くお世話になりました。そして川崎賢弥氏には本報告の最初から最後まで熱心かつ親身なご指導を頂きました。御三方には厚く感謝致します。最後に本報告の作成にあたり精神的に支援を頂きました友人や家族をはじめ，村田研究室の皆様に感謝の意を表しまして謝辞とさせていただきます。

参考文献

- [1] 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*, December 2009, pp. 1–6.
- [2] Y. Liu, J. Geurts, J.-C. Point, S. Lederer, B. Rainer, C. Muller, C. Timmerer, and H. Hellwagner, “Dynamic adaptive streaming over CCN: a caching and overhead analysis,” in *Proceedings of 2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3629–3633.
- [3] R. Ravindran, T. Biswas, X. Zhang, A. Chakraborti, and G. Wang, “Information-centric networking based homenet,” in *Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 1102–1108.
- [4] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, “Developing information networking further: From PSIRP to PURSUIT,” in *Broadband Communications, Networks, and Systems*. Springer Berlin Heidelberg, October 2012.
- [5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *Proceedings of 5th international conference on Emerging networking experiments and technologies (CoNEXT 2009)*, December 2009, pp. 1–12.
- [6] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, “Named data networking (NDN) project,” PARC, Tech. Rep. NDN-0001, October 2010. [Online]. Available: <http://named-data.net/wp-content/uploads/TR001ndn-proj.pdf>
- [7] K. Massri and A. Vitaletti, “DTN routing protocols on resource constrained devices: Design, implementation and first experiments,” in *Proceedings of 2013 21st International Conference on Software and Telecommunications and Computer Networks (SoftCOM)*, September 2013, pp. 1–5.
- [8] N. Uchida, N. Kawamura, T. Ishida, and Y. Shibata, “Resilient Network with Autonomous Flight Wireless Nodes based on Delay Tolerant Networks,” *IT Convergence Practice (INPRA)*, vol. 2, no. 3, pp. 1–13, September 2014.

- [9] G. Tyson, J. Bigham, and E. Bodanese, "Towards an Information-Centric Delay-Tolerant Network," in *Proceedings of the IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking*, September 2013.
- [10] Y.-T. Yu, C. Tandiono, X. Li, Y. Lu, M. Sanadidi, and M. Gerla, "ICAN: Information-centric context-aware ad-hoc network," in *Proceedings of 2014 International Conference on Computing and Networking and Communications (ICNC)*, February 2014, pp. 578–582.
- [11] "CCNx," PARC, October 2014. [Online]. Available: <http://www.ccnx.org/>
- [12] "AR.Drone 2.0 Power Edition," Parrot, October 2014. [Online]. Available: <http://ardrone2.parrot.com/>
- [13] "Raspberry Pi B+," Raspberry Pi Foundation, October 2014. [Online]. Available: <http://www.raspberrypi.org/>
- [14] "ARDroneForP5," Engineering Navi, October 2014. [Online]. Available: <http://kougaku-navi.net/ARDroneForP5/>
- [15] "RXTX for Java," RXTX, December 2014. [Online]. Available: http://rxtx.qbang.org/wiki/index.php/Main_Page