

特別研究報告

題目

コンテンツセントリックネットワーク技術による
ストリームデータ配信システムの設計と実装

指導教員

村田 正幸 教授

報告者

川崎 賢弥

平成 26 年 2 月 14 日

大阪大学 基礎工学部 情報科学科

コンテンツセントリックネットワーク技術による
ストリームデータ配信システムの設計と実装

川崎 賢弥

内容梗概

近年、IP ネットワークに代わる新世代ネットワークの登場が期待されている。新世代ネットワークの一つとしてコンテンツセントリックネットワーク (Content Centric Network: CCN) がある。CCN は IP アドレスの代わりにコンテンツ自身の名前を用いてパケットのルーティングを行うことにより、コンテンツ主体の通信形態を実現する通信パラダイムである。これにより、より簡便な資源発見が可能になるだけでなく、中間ノードによるコンテンツキャッシュを実現することでコンテンツの可搬性が向上し、ネットワーク利用効率の向上や通信における応答時間の短縮などが期待される。CCN に関する研究はまだ基礎段階のものが多く、アーキテクチャのみならずアプリケーションについてもさまざまな研究がなされているところである。しかしながらアプリケーションに関する研究の多くは、すべてのコンテンツを独立とみなし、個別のコンテンツ取得を前提としており、ストリームデータなど時間系列により生成される一連のコンテンツを取得する手法はほとんど検討されていない。

本報告では CCN 技術を用いてストリームデータを生成・配信するシステムの設計および実装を行う。ストリームデータの伝送において重要となるランダムアクセス性やセッションの柔軟性を実現しつつ、コンテンツ提供ノードへの制御も含めたシームレスなネーミングアーキテクチャとそれを制御するコンテンツセントリックネットワークにより、特に計算資源の乏しい組み込み機器においても柔軟性の高いコンテンツ配信機構を実現することを目標とする。提案システムの実現可能性について検証するため、組み込み機器プラットフォームを用いた無線センサネットワークにおいてシステムを実装し、動作検証を通じてシステムの有効性を確認する。

主な用語

コンテンツセントリックネットワーク (CCN)、ストリームデータ、ルーティング、資源発見、センサネットワーク

目次

1	はじめに	5
2	関連研究	9
2.1	CCN 技術を用いたアプリケーションに関する研究	9
2.2	資源制約のあるデバイス上における CCN の実現に関する研究	11
3	CCN 技術を用いたストリームデータ配信システムの設計	12
3.1	ストリームデータの定義	12
3.2	ストリームデータのコンテンツ構成	14
3.3	ストリームデータの名前構造	15
3.4	コンテンツの要求方法	16
4	無線センサネットワークにおける CCN を用いたストリームデータ伝送の実装	18
4.1	無線センサネットワーク概要	18
4.2	使用機器	19
4.3	無線センサネットワークの実装方法	21
5	実験ネットワークの動作検証	23
5.1	動作実験の概要	23
5.2	動作実験の環境	23
5.3	動作実験の方法	24
5.4	動作実験の結果	25
6	終わりに	27
	謝辞	28
	参考文献	29

目 次

1	CCN における通信手順	6
2	Interest 集約の例	7
3	NDNVideo で提案している名前空間	10
4	ストリームデータ配信システムの概要	13
5	動画配信におけるストリームデータとコンテンツ	14
6	指定した期間のデータに対する要求時のシーケンス図	17
7	実装する無線センサネットワークの概要	18
8	Armadillo-420 の外観	20
9	カメラを接続した Armadillo-420 の外観	20
10	動作実験環境	24
11	実験で得られた映像フレーム	25

表 目 次

1	NDNVideo で拡張された Interest パケットフォーマット	10
2	ストリームデータの名前構造	15
3	Armadillo-420 のハードウェア仕様	19
4	コンテンツ名の構成	21
5	生成したコンテンツ名	23

1 はじめに

今から 40 年ほど前に誕生したインターネットであるが、当初は機器の遠隔操作やファイル転送などが主たる用途であり、対象ノードが既知である通信がほとんどであった。このため、現在でも広く用いられている IP (Internet Protocol) では、通信先は通信ノードの識別子である IP アドレスを指定して通信を行うように設計されている。しかしながら、インターネットが社会全体に広く普及し、特に World Wide Web の普及と全文検索型サーチエンジンの登場によってインターネットの利用形態は情報やサービスの取得が主となってきた。インターネット利用者は全文検索型サーチエンジンを利用して取得したいコンテンツを保持している通信ノードを検索し、その結果に含まれる通信ノードを指定して通信することで目的のコンテンツを手に入れる。このように現在では、エンドユーザの関心は通信相手ではなくコンテンツ自身へと移行している。しかし、このように利用形態が変化した現在においても依然として IP が通信プロトコルとして用いられており、通信の非効率性やコンテンツの永続性確保などにおいて利用形態と設計指針の乖離による問題が生じている。

このような問題を解決するため、IP に代わる全く新しいネットワークアーキテクチャとしてコンテンツセントリックネットワーク (Content Centric Networking: CCN)、あるいは情報セントリックネットワーク (Information Centric Networking: ICN) が注目されている。CCN はこれまで欧州の PURSUIT [1]、SAIL [2]、COMET [3] や米国の DONA [4]、NDN [5]、CCNx [6]、MobilityFirst [7] などさまざまなプロジェクトによって研究が進められている。本報告では、このうち NDN/CCNx に代表されるコンテンツセントリックネットワークを対象とする。CCN における通信ではコンテンツの受信側から通信を開始し、以下の順序でコンテンツの取得を行う。このときの概略を図 1 に示す。

1. データの受信側は要求するコンテンツ名を指定した Interest パケットをネットワークに送信する。
2. Interest パケットで指定されたコンテンツを持つノードはコンテンツを Data パケットに格納して返信する。
3. Data パケットを返信する際に中継ルータがコンテンツをキャッシュする。
4. 別のユーザから前回と同じコンテンツに対する Interest パケットが送信される。
5. 対応するコンテンツをキャッシュしているルータが応答してコンテンツを返信する。

このように CCN では「コンテンツ」を通信主体とし、コンテンツの名前を経路制御に直接用いることでコンテンツ主体型の通信を実現する。さらにコンテンツを主体とした通信形態をとるためネットワークルータによるコンテンツのキャッシュが可能になり、ユーザの要

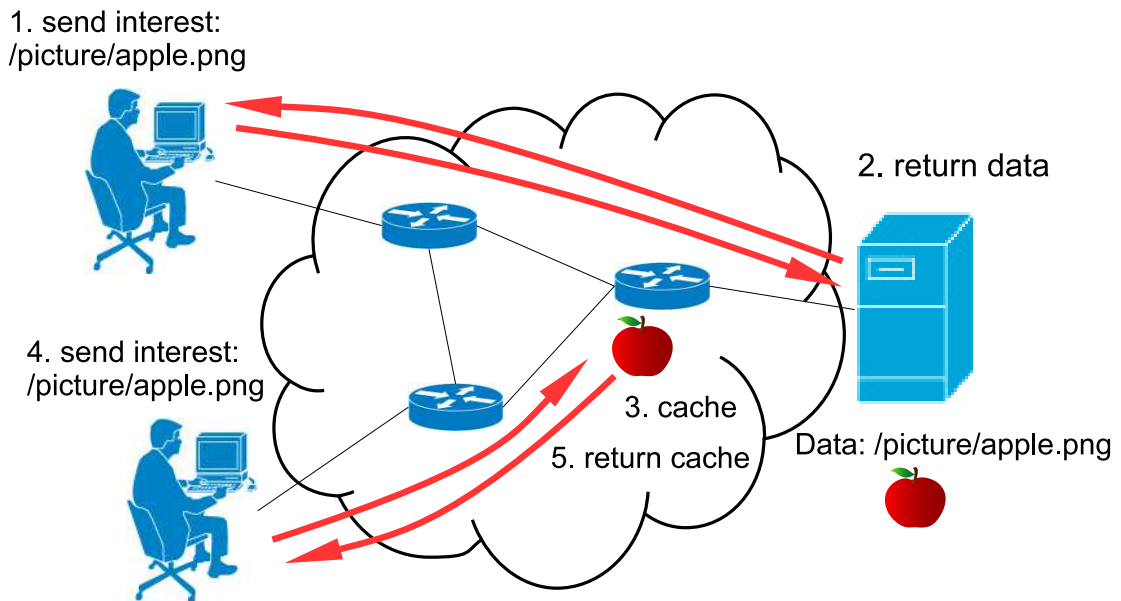


図 1: CCN における通信手順

求にネットワークルータが応答することでネットワーク利用効率の向上や応答時間の短縮などが見込まれる。

このような通信を実現するため、CCN におけるルータは Pending Interest Table (PIT)、Forwarding Information Base (FIB)、Content Store (CS) の 3 つの経路探索テーブルが構成されている。PIT は Interest パケットが送信されてきたインタフェース (通信ポート) を保持するためのテーブルである。PIT は Interest パケットに対応する Data パケットがルータまで返送される時、さらにコンテンツ要求元に Data パケットを転送するために参照される。FIB は送信されてきた Interest パケットに対し、目的のコンテンツを保持するノードへの転送先を管理するテーブルである。FIB は IP ネットワークにおける経路表と同様の構造であるが、IP における経路表では検索キーが IP プレフィックスであるのに対し FIB では検索キーがコンテンツ名 (あるいはその一部) となる点が異なる。FIB のエントリは IP と同様にコンテンツ名集約が行われ、集約された名前に対するプレフィックスマッチングを行うことによって Interest パケットの送信先を決定できる。CS は伝送されてきたコンテンツを当該ルータに一時的にキャッシュしておくための領域である。CS にキャッシュされているコンテンツに対する Interest があつた場合、CS 内のコンテンツを Data として返信することで目的ノードまで到達する前に中間ノードでコンテンツを返すことが可能となり、応答時間の短縮やネットワーク混雑の低減が期待される。CS の領域は有限であるため、キャッシュするコンテンツはルータごとに設定されるキャッシングポリシーに従って決定される。

さらに CCN ではルータによる Interest 集約とマルチキャストをサポートしている。Interest

集約とは、あるルータに同一コンテンツを要求する Interest パケットが複数のインタフェースから到着した場合それぞれのパケット到着インタフェースを PIT に登録し、最初に到着した Interest パケットのみ次のルータへ送信する方式である。図 2 に示すとおり、ルータにおいて複数のインタフェース (face1、face2) に到着した Interest は、そのうち一方のみが次ノードに伝送される。これに対しルータによるマルチキャストとは、Data パケットを受信したときに対応した PIT エントリに登録されている全てのインタフェースに対してパケットを複製して送信する方式である。

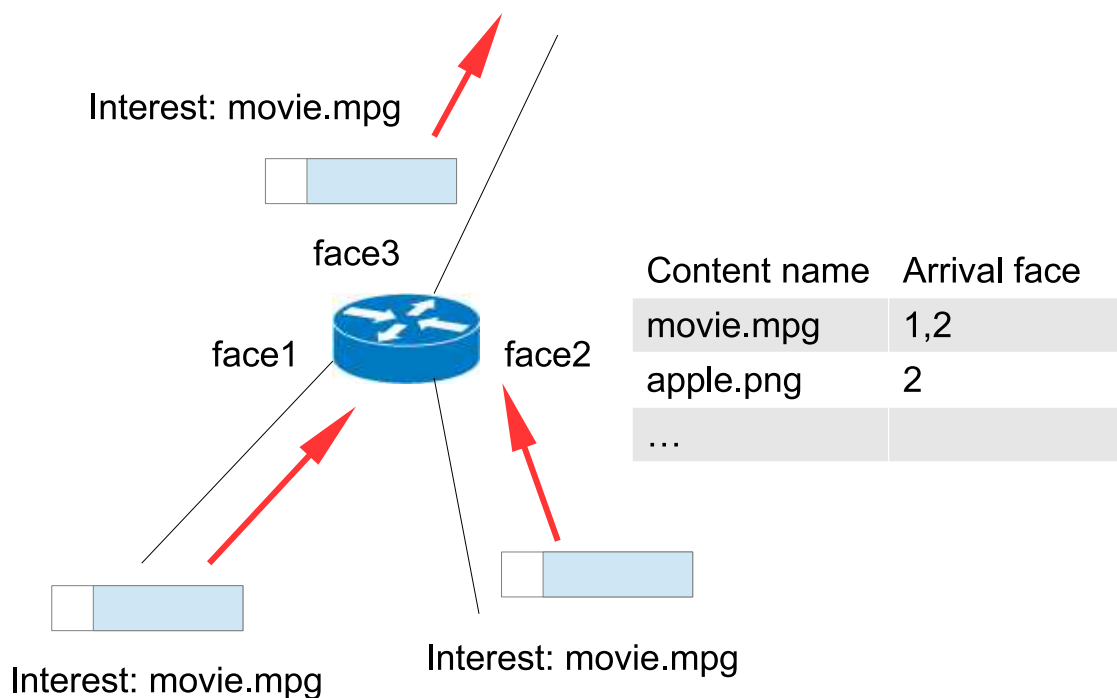


図 2: Interest 集約の例

CCNに関する研究は多くが基礎段階であり、CCNの通信アーキテクチャのみならず、CCNによるアプリケーションに関する研究も盛んに進められている。特に、単なるコンテンツの取得から発展させ、CCNの利点を有効に活用するアプリケーションに対する期待が高まっている。すでに、インターネット電話 [8] やビデオストリーミング [9, 10] などのアプリケーション実装や、アプリケーションを効率よく利用するためのネットワークアーキテクチャ [11] などの研究が行われている。しかしながら、これらのアプリケーション研究の多くはすべてのコンテンツを独立した単独コンテンツとしてその要求・伝送が行われているにすぎない。ストリームデータのように時間系列によって生成される一連のコンテンツ統一的に扱うためには、単なる独立コンテンツの取得だけでは、コンテンツを取得するための制御情報や、時間情報などの指定およびその取り扱いについて考察されておらず、不十分である

と考える。文献 [8] では Voice over IP (VoIP) のセッション管理シーケンスを CCN によって実現する手法が提案されている。しかしながら、この方法は VoIP における SIP (Session Initiation Protocol) のシグナリングシーケンスを単に CCN の Interest 表記に置換しただけであり、CCN における制御特性を考慮したものではない。また文献 [9, 10] などではストリームデータに関する議論が行われているが、ルータにおいて独自の処理が必要である、あるいは各ノードの詳細な動作が定義されていないなどの課題があり、ストリームデータを統一的に扱うためのモデル化が十分でない。

一方 CCN の有力なアプリケーションネットワークとして、IoT (Internet of Things)、M2M (Machine-to-Machine)、あるいは家電ネットワークなどが挙げられる。これらのネットワークでは、各ノードは計算資源が乏しい組み込み機器であることも十分に想定され、通信のインテリジェンスを実現するためにすべてのノードに負荷の高い処理を求めることは容易ではない。以上の背景から、組み込み機器などから構成されるネットワークにおける CCN の実現に関する検討がなされている [12, 13] が、ここでもコンテンツは独立なものとして考えられており、ストリームデータの扱いについては詳細に検討されていない。

本報告では、CCN においてストリームデータを統一的に扱うためのアーキテクチャ、および要素技術に関する検討を行い、ストリームデータ配信システムの設計および実装を行う。まず、ストリームデータの統一的なモデルを提案し、それを CCN 上で実現するためのネーミングアーキテクチャ、および各ノードの動作について考察する。さらに、動作シーケンスについて詳細に設計する。この際、ストリームデータの取得において重要となるランダムアクセス性やセッションの柔軟性を CCN においてどのように扱うかを検討する。

以上の提案システムの実現可能性を検証するため、組み込み機器で構成されるセンサネットワークにおいて提案システムを実装し、動作検証によってその有効性を確認する。

2 関連研究

本章では CCN 技術を用いたアプリケーションに関する研究と、資源制約のあるデバイスにおける CCN の実現に関する研究について述べる。

2.1 CCN 技術を用いたアプリケーションに関する研究

CCN 技術を用いたアプリケーションは現在活発に研究されており、例えば CCNxCon [14] では、CCNx を用いたユースケースやアプリケーションなどが提案されている。ストリームデータの一つである音声通話については、文献 [8] において VoIP (Voice over IP) の CCN による実現手法が提案されている。

また文献 [9] では、CCN によるビデオおよび音声ストリーミングアプリケーションを実現するための NDNVideo が提案されており、プロトコルおよびデータシーケンスが提案されている。[9] では、ユーザの要求に応じた多様なストリーミングメディアを提供するため、図 3 で示される、階層型の名前空間を提案している。

NDNVideo での名前構造は、コンテンツ自身を根とする木構造で構成されており、コンテンツは複数の属性を持つサブコンテンツの集合として表現されている。また、コンテンツのメディア種類（ビットレート、コーデック、セグメント等）が子ノードにおいて定義されており、ユーザは所望する品質のコンテンツに関する情報を Interest パケットのコンテンツ名として指定することで、ランダムアクセスや動的な品質調整が可能なストリーミングシステムが実現可能である。

さらにデータのランダムアクセスを実現するための Interest パケットの構成と通信シーケンスについて提案されている。ここでは、インデックスに対する Interest 要求のパケットに 3 つの要素の追加が提案されている。一つ目は、ChildSelector という属性であり、この属性を RIGHTMOST に設定するとインデックスの名前空間のうち最後の要素を要求することを示す。二つ目は、Exclude という属性であり、この属性に指定した期間のデータは要求していないということを示す。三つ目は、AnswerOriginKind という属性であり、この値を NONE に設定することでローカルのルータにキャッシュされているコンテンツを無視してオリジナルのコンテンツを問い合わせるというものである。これらを設定した Interest パケットの例を表 1 に示す。

表 1 で示すパケットが送信された場合、00:00:05:01 以降のデータを除き最も最新のインデックスをネットワークに要求し、コンテンツとして 00:00:05:00 のデータが返信されてくることを期待している。しかしながら実際には各中継ルータが返送するキャッシュコンテンツは要求情報を満たすが期待したデータではない可能性がある。例えば中継ルータが 00:00:04:00

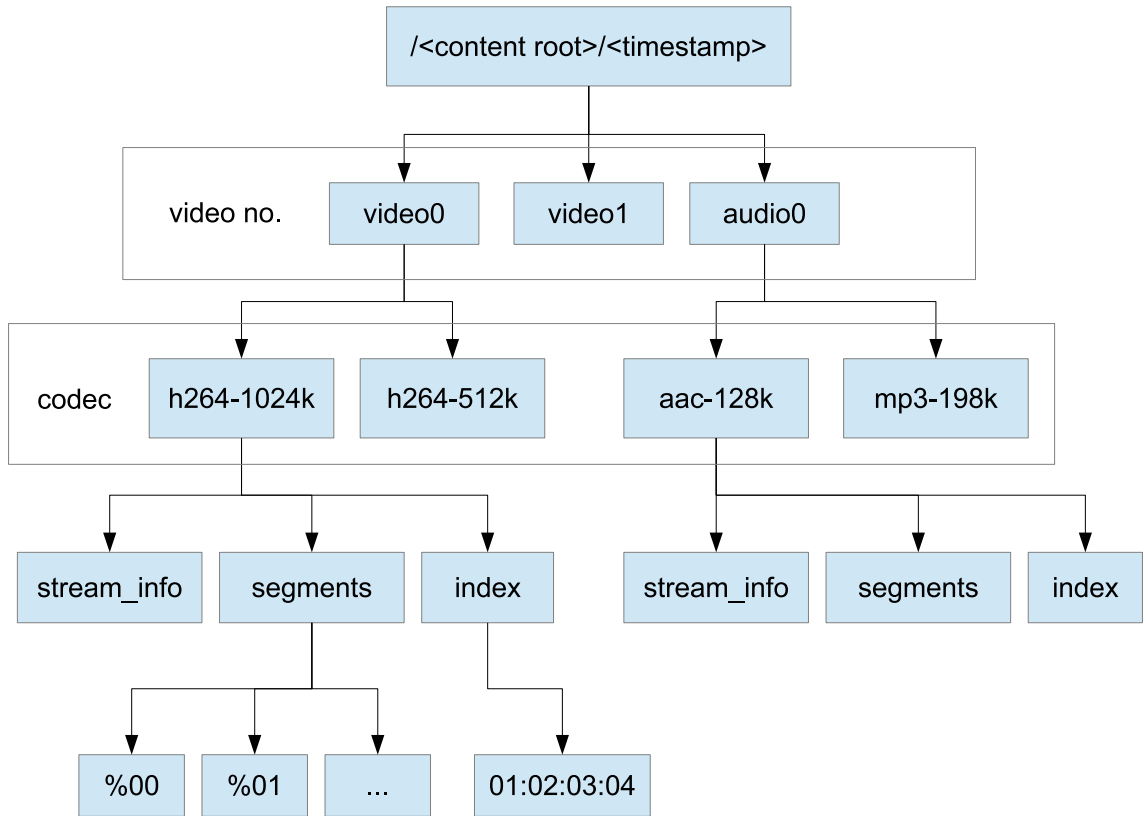


図 3: NDNVideo で提案している名前空間

表 1: NDNVideo で拡張された Interest パケットフォーマット

Name: ../index	
AnswerOriginKind: NONE	
ChildSelector: RIGHTMOST	
Exclude:	00:00:05:01 ANY

のデータを保持していた場合、このデータは要求情報を満たすためユーザに返送されてしまう。ユーザが正しくコンテンツを受信するためにはストリーミングデータについてパケット単位の状況を把握し、希望しないコンテンツデータを Exclude で指定する必要がある。加えて、表 5 の属性を各ルータで実現するためには、ストリーミングための処理をすべてのルータに対して追加する必要がある、段階的な移行ができないという問題がある。

2.2 資源制約のあるデバイス上における CCN の実現に関する研究

すでに述べたとおり、CCN は特に IoT や M2M、家電ネットワーク等で有用であると期待されているため、CCN を特に計算資源の制約が厳しいデバイス上で実現するための方法について研究が進められている [10, 12, 13]。

文献 [10] では Android 携帯端末上でビデオストリーミングを行う際のコンテンツの扱いについて少し触れている。しかしストリームデータの扱いに関しては、ストリームごとにセグメントや品質に関するメタデータを用意すること、コンテンツ名は検索エンジンかメタデータによって探索できることの二つについてのみ述べられている程度に留まっており、詳細な設計がなされていない。

文献 [12] では組み込みシステム向け OS である Contiki に CCN 通信モジュールを統合し、シミュレーションと実機による実験を行っている。シミュレーションによる実験ではネットワーク規模を変化させた場合において、単一のコンテンツに対して Interest を送信した場合に応答が得られるまでの遅延時間を評価している。実機による実験では、キャッシュを有効化した場合と無効化した場合においてコンテンツ受信にかかる遅延時間の変化について調査している。

文献 [13] ではネットワーク中のノードを計算資源の制約あり、なしで分類し、階層型のネットワーク構造により連携を行う方式が提案されている。具体的には資源の制約が厳しいネットワークデバイスは、計算資源の制約のないノードで構成されるコアネットワーク中のスーパーノードに対して Interest 要求を送信することで、Interest を受信したスーパーノードが改めて Interest 要求を出すという入れ子構造の Interest-Data 交換を提案している。ネットワークデバイスから送信する Interest にはコンテンツ自身ではなくスーパーノードが提供しているサービスの名前を指定する。このように資源の少ないノードはスーパーノードをプロキシとして利用することで、ノードが処理できないタスクをネットワーク上で処理することができる。処理されたデータは最初に送信した Interest に対する返信として Interest 送信ノードで受信される。

3 CCN 技術を用いたストリームデータ配信システムの設計

本章では CCN 技術を用いてストリームデータを配信するさまざまなアプリケーションに適用できる一般化したシステムの設計を行う。はじめに、ストリームデータの定義とそのモデル化について行い、次にそのモデルを CCN で実現するための必要な個別要素について検討する。具体的には、ストリームデータのコンテンツ構成および名前構造、コンテンツの要求方法について議論する。

3.1 ストリームデータの定義

本報告においてストリームデータとは、単一ソース（コンテンツ配信元）から時間系列に沿って生成される一連のコンテンツシーケンスであると定義する。ここで、コンテンツ配信元であるソースは物理的に単一である場合だけでなく、論理的（仮想的）なソースを定義することで複数の物理資源が実質的なソースとなる場合もある。

ストリームデータの代表例としては、ビデオや音声などの一般的なストリーミングデータ、定期的にモニタリングされるセンサからの観測データなどが挙げられる。

ストリームデータの通信では、まずコンテンツに含むデータの単位を定める必要がある。例えば動画データをコンテンツ化する場合を考えると、動画データ全体を一つのコンテンツとして扱う方法や、転送可能な最大セグメントサイズで分割してコンテンツにする方法、時間間隔で分割してコンテンツにする方法などが挙げられる。データの単位によって Interest/Data パケットの交換頻度やキャッシュ利用効率などの性能が変化するため、アプリケーションやネットワークの特性によって決定する必要がある。

次にストリームデータを指定するためのネーミングアーキテクチャについて検討する必要がある。CCN ではコンテンツに付ける名前の規則は定義されておらず、コンテンツの発行者が自由に名前を付けることができる。そのため、アプリケーションやユーザにとって可用性の高いコンテンツ名を付けることは重要な課題である。

さらにストリームデータの通信において有用となる、任意の時刻のデータから受信を開始できるランダムアクセス性や、データ受信者の要求に応じて動的に動画のビットレートや音声のサンプリングレートなどデータの品質に関する変更ができる能力などが挙げられる。ランダムアクセス性が実現できれば、受信者にとって必要なデータだけ取得することが可能になり利便性が向上する。また、データの品質に関する変更を行う能力が実現されていれば、ネットワークの輻輳や受信者側の何らかの理由によりコンテンツのデータ品質を下げるなどの要求に対応することが可能になる。以下の各節では、上述の性質を実現するために CCN において検討すべき項目として、ストリームデータのコンテンツ構成および名前構造、送受信シーケンスについて詳細に述べる。

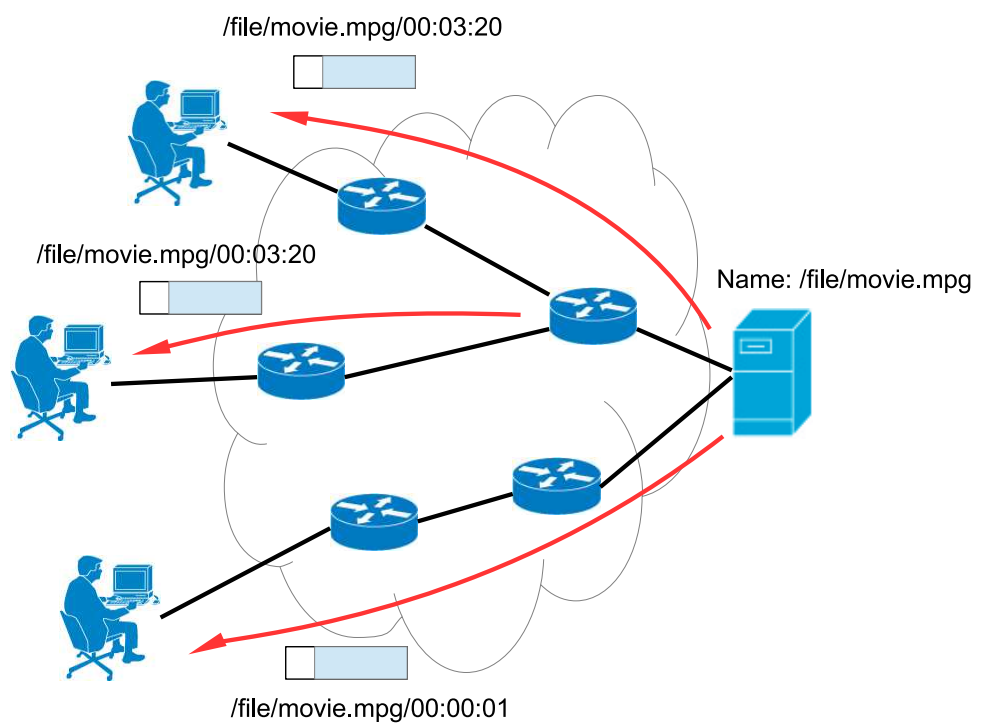


図 4: ストリームデータ配信システムの概要

3.2 ストリームデータのコンテンツ構成

コンテンツに含むデータの単位はいくつか挙げられるが、本報告ではストリームデータが持つ意味的な区切りに従ってデータをコンテンツに分割する方法を提案する。ここでいう意味的な区切りというのは、例えば動画データならフレーム単位、音声データならサンプル単位、センサのログファイルではデータが生成される単位時間など、単独で意味を持つデータとして取得可能な最小のデータサイズを提供するために必要となる時間間隔を指す。動画データの場合におけるコンテンツ作成例を図5に示す。

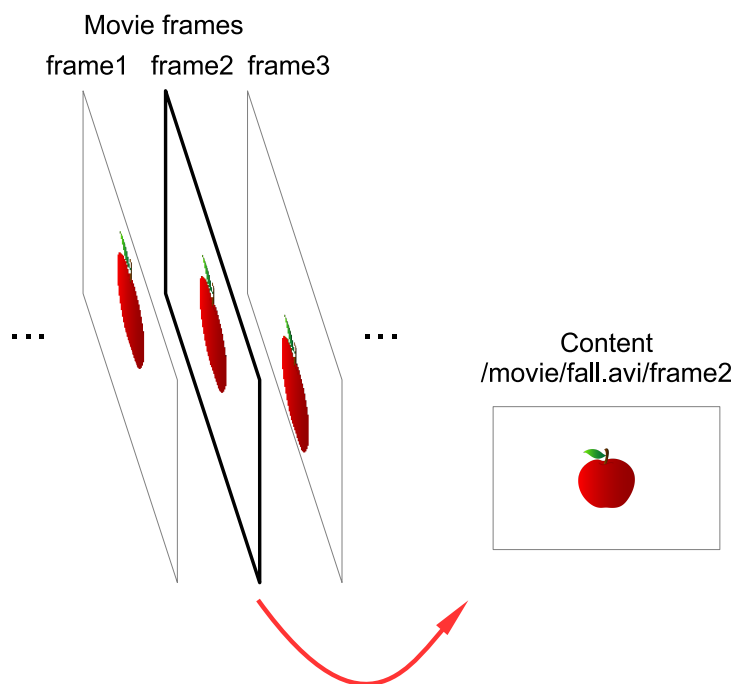


図 5: 動画配信におけるストリームデータとコンテンツ

このような分割を行うことの利点として、ランダムアクセス性や動的なデータクオリティ変更の実現が容易になるということが挙げられる。コンテンツごとに一意な名前付けを行うことができるため、そのコンテンツに含まれているデータがストリームデータのうちのどの位置付けにあるのかという情報をコンテンツ名に反映することが容易である。それに加え、同じストリームデータからクオリティを変えた複数のコンテンツを作成しておくことで、クオリティ変更要求が行われた際には要求するコンテンツを変更するだけで同じ位置から変更されたクオリティでデータを取得することが可能である。コンテンツ全体を一つのコンテンツとして扱う場合やデータサイズで分割してコンテンツ化した場合でも同様の性質の実現は可能であるが、コンテンツをパッケージ化する際に分割されるセグメント単位とストリームデータ内の位置との間で変換が必要となってしまう。

3.3 ストリームデータの名前構造

CCN では名前によるルーティングが可能であるため、名前構造の決定はアプリケーションの本質的な性質を決めることとなる。コンテンツ名が果たす役割としては、ルーティング情報であるという他にそのコンテンツが持つ情報を的確に示していることが望まれる。ストリームデータの通信を実現するためのコンテンツとメタデータに対するネーミング規則として表 2 で示す構造を提案し、それぞれの要素について以下に考察を行う。

表 2: ストリームデータの名前構造

Content type	naming
Content	/<routing prefix>/<data name>/<version>/<quality>/<sequence no>
Metadata	/<routing prefix>/<data name>/metadata

プレフィックス (routing prefix)

プレフィックスは CCN におけるルーティングを効率化するために用いる名前である。CCN におけるルーティング情報は IP で行われているのと同様に集約可能でなければ、FIB の爆発的な増大を抑制することができない。特に、CCN の場合事実上無制限に名前を生成することが可能であるため、コンテンツ名が集約可能である構造にすることは、CCN ルータにおける経路表の爆発を抑える上でも重要である。コンテンツ名の集約を考える場合、コンテンツが配置される位置に依存する名前をコンテンツ名の先頭に置くのが望ましいと考えられる。

ストリームデータ名およびバージョン (data name, version)

ストリームデータ名はストリームデータを一意に定めるためにストリームデータ配信者が付ける名前である。少なくとも同一プレフィックス内では一意性を求められることから、名前空間を導入する必要もあると考えられる。

また、コンテンツの更新や置き換えを行う場合に元のコンテンツと区別するためにバージョン情報を挿入する。この情報はコンテンツが生成された日時をベースとした名前にすることで元のコンテンツのバージョン情報との競合を避ける。

データ品質 (quality)

データ品質はストリームデータの再生・復元を行う際に必要なデータ品質の情報を表現する名前である。この部分で表現する要素としては、ファイルの形式であったり動画データであればビットレートや解像度など、音声データであればサンプリングレートなどが挙げられる。コンテンツ配信者は予めそれぞれの品質で生成したデータからコンテンツを作成し、配信可能な状態にしておく。受信者は自身が要求する設定を Interest の名前で指定してコンテンツを要求することで、希望のデータ品質でコンテンツを受信することができる。

シーケンス番号 (sequence no)

シーケンス番号は受信するコンテンツに含まれるデータの順序を表す情報である。この部分は一連の番号をベースとして名付けられており、次に取得すべきコンテンツの名前が予測できる形になっていなければならない。一連の番号として用いるものは自由であり、単純にデータのセグメント番号であったり、動画であればフレーム番号、音声であればサンプリング番号、データが等しい時間間隔で生成されるものであれば生成時刻を用いてもよい。

メタデータ (metadata)

通信を始める際、コンテンツ名のうち通信設定以降の部分に関して有効な名前というのは受信者にとって未知のものである。どのような名前に対する要求が有効であるのか知るために、メタデータの活用が有効であると考えられる。配信者が提供する準備ができていない通信設定やデータの存在する時刻の範囲などを表すコンテンツをメタデータとして用意し、ルーティングプレフィックスとコンテンツに含まれるデータの内容を表す名前の直後に “metadata” などの予め定義しておいたキーワードを続けることで受信者にとって既知の名前のみでメタデータへのアクセスが可能となる。

3.4 コンテンツの要求方法

任意の期間においてストリームデータを受信しようとした場合、単純な方法として次の二つが考えられる。一つは、ストリームデータの受信開始と終了の際にそれぞれ Interest パケットを送信し、配信者は終了パケットが到着するまで次々とコンテンツを送信し続けるという方法である。もう一つは、ストリームデータから作成したコンテンツそれぞれに対して Interest パケットを送信するという方法である。前者の方法では Interest パケットを2回送信するだけでよいのでネットワークの帯域を節約することができる。しかし、パケット損失

時の再送処理が複雑になることや、CCN の設計原理として述べられている“ホップ間のフローバランスを調整するために一つの Interest パケットには一つの Data パケットが対応する” [5] という原理に沿わないことが欠点である。後者の方法では Interest パケットの送信回数は増えてしまうが、パケット損失時の再送には該当する Interest パケットを再送するだけで良く、フローバランスの調整も行われている。さらにこの方法ではストリームデータの一部分のスナップショットを取得する際に特別な動作を必要としない。そのため本報告では後者の方法を採用することとする。

ストリームデータを任意の期間にわたって取得する場合の動作シーケンス図を図 6 に示す。まず、取得したいコンテンツ名を指定した Interest パケットを送信する。(図中①) 次に Interest で指定されたコンテンツを持つ機器は、対応するコンテンツを返信する。(図中②) 要求したコンテンツを受信したのち、続けて次のデータを取得する場合にはコンテンツのバージョンを一つ先に進めた Interest パケットを送信する。(図中③) ストリームデータの取得を終了するまでの間この Interest パケットとコンテンツの交換を繰り返す。

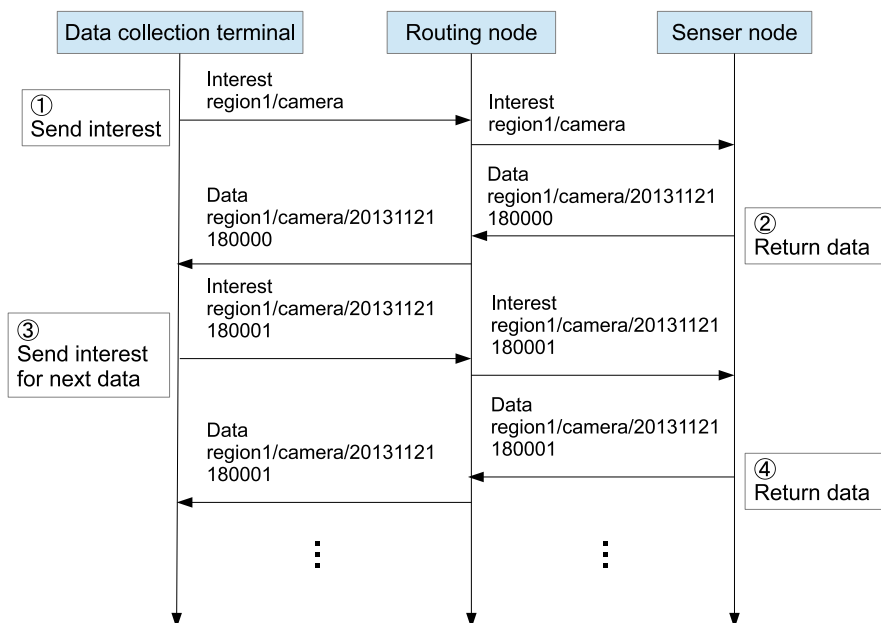


図 6: 指定した期間のデータに対する要求時のシーケンス図

4 無線センサネットワークにおける CCN を用いたストリームデータ伝送の実装

4.1 無線センサネットワーク概要

本報告で実装する無線センサネットワークの概要を図7に示す。実装する無線センサネットワークはノードとセンサ、データ収集用端末によって構成される。センサとしてはストリームデータを発生させるものとしてカメラを用い、撮影した映像をセンシング情報として扱うこととする。ノードはセンサを取り付けたセンシングノードとパケットのルーティングを行うルーティングノードの二種類に分けられる。センシングノードは取り付けられたカメラから得られる映像から常時コンテンツを生成し、コンテンツの配信者として動作する。ルーティングノードは Interest パケットや Data パケットのルーティングを行う。データ収集用端末はセンシングノードからのセンサ情報を収集するものであり、ストリームデータの受信者として動作する。

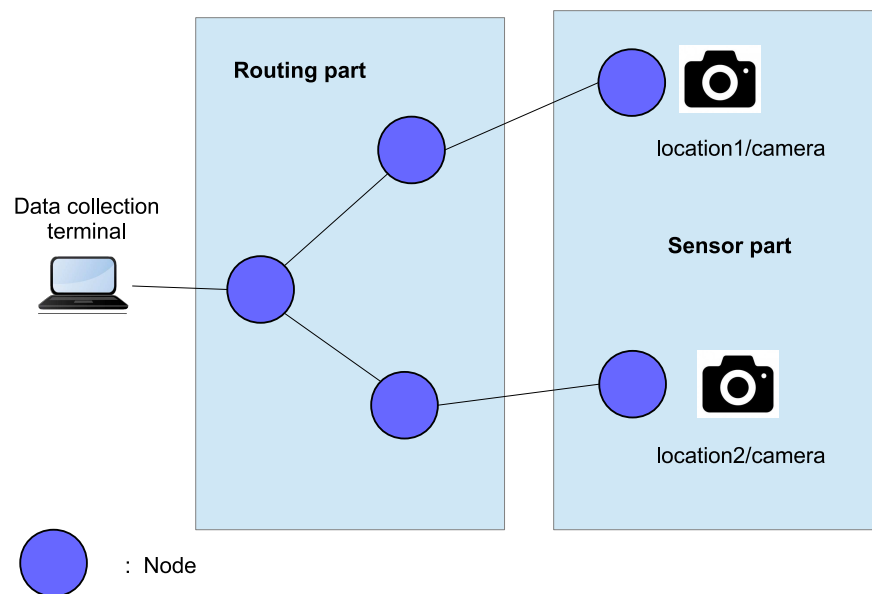


図 7: 実装する無線センサネットワークの概要

今回の実装ではノードとして組み込み機器プラットフォームを用いた環境においてストリームデータの配信が行えるのか確認することが目的のため、センサが取得したデータを蓄積しておくストレージは実装せず、データ収集用端末からセンサが収集している最新のデー

タを取得することのみ行う。

4.2 使用機器

本報告ではノードとして組み込み機器プラットフォームである Atmark-techno 社製の Armadillo-420 を使用した。Armadillo-420 の外観を図 8 に示す。Armadillo-420 はクロックが 400MHz の CPU を搭載し、64MB の LPDDR SDRAM と 16MB のフラッシュメモリを備えた小型の CPU ボードである。Linux2.6 がインストールされており、シリアルポートや LAN ポート、USB 入力端子、microSD スロット、GPIO などの入出力機能を有している。これに Armadillo-WLAN モジュールを取り付けることにより IEEE 802.11b/g/n 規格による無線 LAN 通信が可能となる。Armadillo-420 のハードウェア仕様について表 3 に示す。

表 3: Armadillo-420 のハードウェア仕様

プロセッサ	Freescale i.MX257
CPU コア	ARM926EJ-S
CPU コアクロック	400MHz
バスクロック	133MHz
RAM	64MB (LPDDR SDRAM)
フラッシュメモリ	16MB (NOR 型)
無線 LAN	IEEE 802.11b/g/n 対応 (最大 72.2Mbps)
USB	USB2.0×2 (High Speed×1, Full Speed×1)

Armadillo に取り付けるカメラには Buffalo 社製の USB 接続 Web カメラを用い、mjpg_streamer というソフトウェアによってカメラからの映像を取得する。カメラを接続した Armadillo を図 9 に示す。mjpg_streamer では HTTP 経由で独自転送方式を用いたビデオストリーミングを行う機能があるほか、Motion JPEG 形式の動画に必要な JPEG 形式の画像フレームを生成することができる。

また、CCN の実装には PARC で開発された CCNx プロトコル [15] を用いた。CCNx プロトコルは IP ネットワーク上に CCN 通信を行うオーバーレイネットワークを構築する形で実装されている。そのため、物理ネットワーク上に任意のトポロジ構成で CCN ネットワークを構築できるようになっている。



図 8: Armadillo-420 の外観



図 9: カメラを接続した Armadillo-420 の外観

4.3 無線センサネットワークの実装方法

センシングノードでは `mjpg_streamer` を動作させ接続されたカメラからの映像を `jpg` 形式の画像フレームとして取得し、フレームごとにコンテンツ化を行う。その際コンテンツに付ける名前の構成を表 4 に示す。表 4 の各要素は 3.3 節で設計したコンテンツのネーミング規則と対応している。

表 4: コンテンツ名の構成

Content type	naming
Content	<code>/<routing prefix>/<location name>/<sensor type>/<version>/<data format>/<frame number></code>
Metadata (format)	<code>/<routing prefix>/<location name>/<sensor type>/metadata</code>
Metadata (frame)	<code>/<routing prefix>/<location name>/<sensor type>/<version>/<data format>/metadata</code>

プレフィックス (routing prefix および location name)

`routing prefix` と `location name` は 3.3 節で提案したプレフィックスと対応している。`routing prefix` には `ccnx:/osaka-u.ac.jp` という固定の名前を用いた。`location` にはセンシングノードを設置する場所の名前を挿入し、この名前によって画像フレームが撮影された位置を表現している。

ストリームデータ名 (sensor type)

`sensor type` は 3.3 節で提案したストリームデータ名と対応している。データの内容を表す名前としてデータを取得しているセンサの種類、今回はカメラを用いているので `camera` という名前を用いた。

バージョン (version)

`version` は 3.3 節で提案したバージョンとそのまま対応している。これはセンサデータのバージョン情報を表すものであり、センシングを開始した日時が挿入される。センシングを行っている間は同一バージョンの新しいコンテンツを生成していると捉えることができ、センシングを中断、再開したときに新規バージョンのコンテンツを生成開始したと考える。

データ品質 (data format)

data format が 3.3 節で提案したデータ品質と対応している。この部分でコンテンツに含まれる画像データのフォーマットを表現している。mjpg_streamer では解像度とフレームレートを指定して jpg 画像を生成できたため、この部分では解像度、フレームレート、データ形式の 3 つを表現した。組み込み機器 Armadillo をノードとしているため今回の実装では解像度が QSIF (176×144)、フレームレートが 1 [frame/sec]、データ形式は jpg のコンテンツ 1 種類のみを生成する。

シーケンス番号 (frame number)

最後に frame number が 3.3 節で提案したシーケンス番号と対応している。この部分では画像データのフレーム番号を 9 桁の連番整数によって表現し、各フレームの順序を表している。

メタデータ (metadata)

実装した無線センサネットワークにおいて今回はセンサを配置した位置とセンサ種類については既知のものであるとしており、バージョン情報以降のコンテンツ名については未知であるとしている。そのため完全なコンテンツ名を構成できるようにするため 2 種類のメタデータを用意した。一つはコンテンツの最新バージョンとデータのフォーマットを取得するためのメタデータである。そしてもう一つはコンテンツの最新フレーム番号を取得するためのメタデータである。今回の実装ではデータのフォーマットが 1 種類しかないため一つのメタデータでこれらの情報全てを含めることが可能であるが、実際にはフォーマットごとに最新のフレーム番号が異なるためメタデータを 2 種類に分割した。

5 実験ネットワークの動作検証

本章では4章で実装した無線センサネットワークが適切に動作することを確認するため、カメラで撮影した映像をいくつかのシナリオに従って取得する実験を行った。以下に実験の概要、環境、方法、結果について示す。

5.1 動作実験の概要

本実験は、設計したシステムを用いたアプリケーションが組み込み機器のような資源制約の大きなデバイス上において適切に動作することを示すために実施する。本実験では複数台の Armadillo と PC を無線で接続し、その上で CCNx のオーバーレイネットワークを構築する。2台の Armadillo にはカメラを接続し、それぞれ撮影した映像からコンテンツを生成する。一方のカメラが撮影している映像を要求し、適切な映像データが返信されてくることを確認することによって、最新フレームのコンテンツ名を探索可能であるというランダムアクセス性の確認を行う。次に、もう一方のカメラが撮影している映像を要求し、返信されてくる映像データがシームレスに切り替えられることを確認することで、動的なコンテンツのクオリティ変更能力の確認を行う。

5.2 動作実験の環境

動作実験は図 10 のような環境で行った。大阪大学 大学院情報科学研究科 A 棟 6 階の A610 研究室と廊下に Armadillo を 3 台配置し、3 台の Armadillo とデータ収集用端末のそれぞれをアドホックモードで無線接続した。3 台の Armadillo のうち 2 台にカメラを接続してセンシングノードとし、残りの 1 台はセンシングノードとデータ収集用端末の間を接続するルーティングノードとした。2 台のセンシングノードで生成するコンテンツ名はそれぞれ表 5 のように定めた。表 5 のコンテンツ名によりカメラから生成するデータは JPG 形式で解像度は QSIF (176×144)、フレームレートは 1 [frame/sec] と指定している。また、コンテンツのバージョン情報はデータの配信を開始した日時を用いて表現している。CCNx のオーバーレイネットワークは図 10 に示すトポロジになるように各リンクを設定した。

表 5: 生成したコンテンツ名

Sensor location	Content name
A6F	ccnx:/osaka-u.ac.jp/A6F/camera/<version no. >/jpg/QSIF/1/<frame number >
A610	ccnx:/osaka-u.ac.jp/A610/camera/<version no. >/jpg/QSIF/1/<frame number >

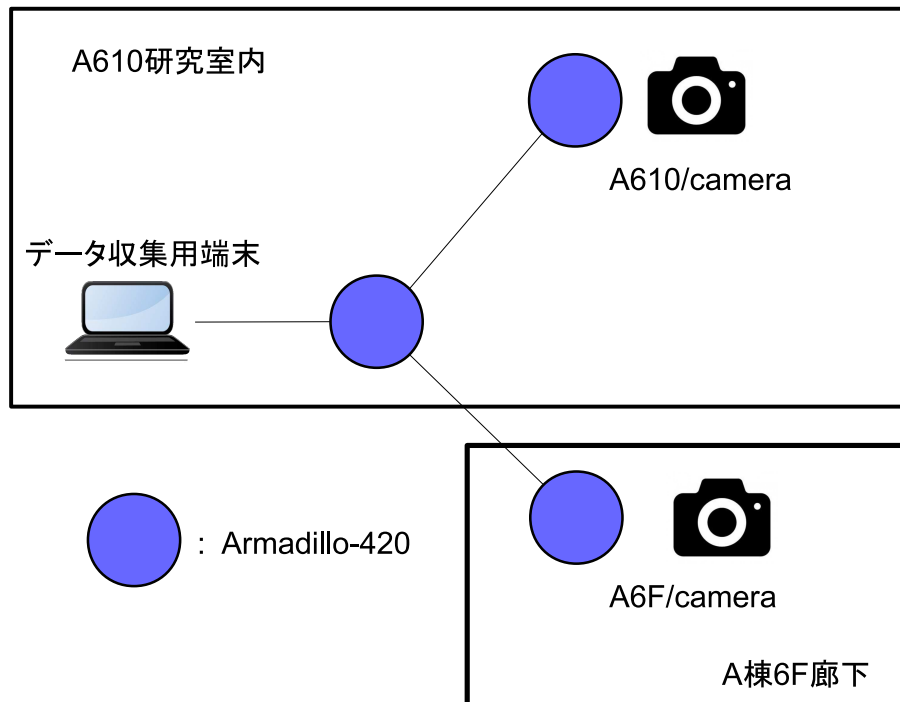


図 10: 動作実験環境

5.3 動作実験の方法

本節では実験の詳細な手順を示す。始めにデータ配信側の準備を行う。まずカメラを接続した各 Armadillo において、コンテンツのバージョン情報、データ形式、解像度、フレームレートを示したメタデータを生成し、“ccnx:/osaka-u.ac.jp/<location name>/camera/metadata” という名前のコンテンツを作成する。次に撮影データの各フレームに対して表 5 に示したコンテンツ名を用いてコンテンツを作成する。このとき最新フレーム番号を示したメタデータを生成し、“ccnx:/osaka-u.ac.jp/<location name>/camera/<version no.>/jpg/QSIF/1/metadata” という名前のコンテンツを作成する。以上の手順を行うことでデータ配信側の準備が完了する。

次に、A 棟 6 階廊下に設置したカメラで撮影した映像データの受信を行う。まず、生成しているコンテンツのバージョン情報や指定可能なデータ形式などの情報を取得するために “ccnx:/osaka-u.ac.jp/A6F/camera/metadata” という名前のコンテンツを要求する。得られた情報からコンテンツ名を構成し、現在の最新フレーム番号を取得するために “ccnx:/osaka-u.ac.jp/A6F/camera/<version no.>/jpg/QSIF/1/metadata” という名前のコンテンツを要求する。得られたフレーム番号を用いてコンテンツの要求を行うことにより現在の映像フレームを取得する。映像フレームの取得が完了するたびにフレーム番号を 1 つずつ進めて

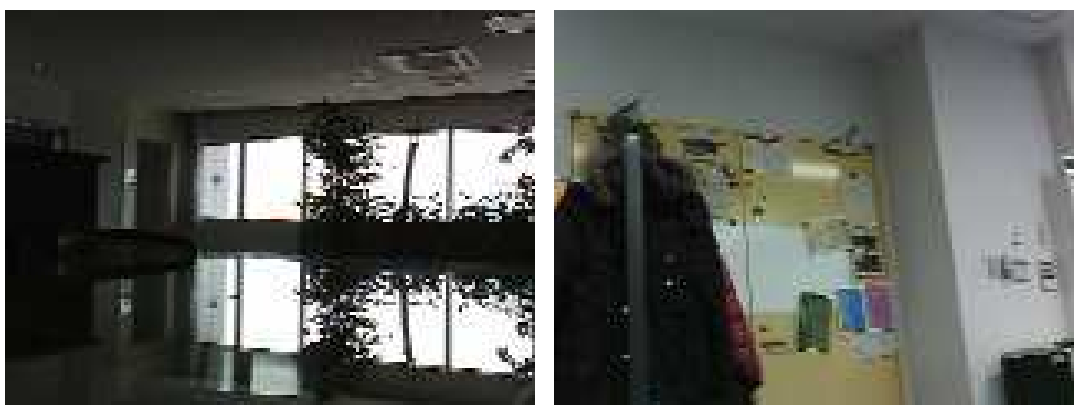
次のフレームのコンテンツを要求することにより、現在カメラが撮影している映像が取得できていることを確認する。これらの手順により、最新フレームのコンテンツ名が探索可能でありランダムアクセス性が実現されていることが確認できる。今回実装した無線センサネットワークではストレージを用意していないため過去のデータを受信することはできないが、ストレージを実装した場合は、バージョン情報に含まれる配信開始日時、フレームレート、最新フレーム番号の3つの情報を用いることで任意の時刻に生成されたフレームを探索することが可能である。

次に、これまでの操作に続けて A610 研究室内に設置したカメラで撮影した映像データの受信を行う。A 棟 6 階廊下に設置したカメラから映像データを取得した場合と同様の操作を行うことにより、受信されるデータが A610 研究室内の映像に切り替わることを確認する。

5.4 動作実験の結果

2 台のカメラで得られたそれぞれの映像のフレームを図 11 に示す。ccnx:/osaka-u.ac.jp/A6F/camera/2014-02-04-14-20-40/jpg/QSIF/1/000000025 というコンテンツ名を要求したところ、A 棟 6 階廊下のカメラから図 11(a) のような jpg 形式の映像フレームが受信されることを確認でき、シーケンス番号を増加させていくことで次の時刻のフレームが継続して受信されることを確認した。

要求するコンテンツ名を A610 研究室内のカメラで撮影しているデータである ccnx:/osaka-u.ac.jp/A610/camera/2014-02-04-14-21-28/jpg/QSIF/1/000000040 に変更したところ、次の受信フレームから図 11(b) のように先ほどとは異なるカメラで撮影された jpg 形式の映像フレームが受信されることを確認した。



(a) A 棟 6 階廊下のカメラから得られたフレーム (b) A610 研究室内のカメラから得られたフレーム

図 11: 実験で得られた映像フレーム

これらの結果から、本報告で設計したシステムを適用したアプリケーションでは最新フレームを探索してアクセスする能力を持ち、要求するコンテンツ名を変化させるだけで受信するストリームを動的に変化させることが可能であることが分かった。

実験中、生成される映像のフレームレートが 0.25 [frame/sec] 程度の速度しか出ていないという現象が発生した。転送に関わる処理にかかる時間を調べたところ、CCNx のソフトウェアルータが行う転送処理がボトルネックとなっていることが分かった。そのため、この現象は低速な組み込み機器を用いたことによる影響であると考えられる。

6 終わりに

本報告では CCN 技術を用いたアプリケーションに適用可能なストリームデータ配信システムの設計を行った。設計においてはストリームデータに対するランダムアクセス性やストリームデータの品質を取得中に動的に変化させる能力を実現するための一般的なシステムアーキテクチャについて議論を行った。そして設計したシステムに基づいて無線センサネットワークを実装し、センサとして接続したカメラから動画の各フレームが連続して取得できることを示した。さらに、要求コンテンツ名のみ変更することで動的に取得フレームを変更することが可能であることを示した。

今後の課題としては、無線センサネットワークにおけるデータ蓄積用ストレージの実装や規模の拡大を行った環境における通信能力の評価や複数のデータ収集端末が存在する場合の評価を行うことなどが挙げられる。

謝辞

本報告を終えるにあたり、様々なご指導を頂きました大阪大学 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、適切なお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、および大阪大学 大学院経済学研究科の小南大智助教に心より御礼申し上げます。最後に、本報告に関して多くの助言を頂きました大岡睦氏、北出雄麻氏をはじめ村田研究室の皆様感謝致します。

参考文献

- [1] N. Fotiou, P. Nikander, D. Trossen, and G. Polyzos, “Developing information networking further: From PSIRP to PURSUIT,” in *Broadband Communications, Networks, and Systems*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, I. Tomkos, C. Bouras, G. Ellinas, P. Demestichas, and P. Sinha, Eds. Springer Berlin Heidelberg, 2012, vol. 66, pp. 1–13.
- [2] B. Ahlgren, P. Aranda, P. Chemouil, S. Oueslati, L. Correia, H. Karl, M. Sollner, and A. Welin, “Content, connectivity, and cloud: ingredients for the network of the future,” *IEEE Communications Magazine*, vol. 49, no. 7, pp. 62–70, July 2011.
- [3] G. Garcia, A. Beben, F. Ramon, A. Maeso, I. Psaras, G. Pavlou, N. Wang, J. Sliwinski, S. Spirou, S. Soursos, and E. Hadjioannou, “COMET: Content mediator architecture for content-aware networks,” in *Proceedings of Future Network & Mobile Summit 2011*, June 2011, pp. 1–8.
- [4] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 181–192, October 2007.
- [5] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, “Named Data Networking (NDN) Project,” NDN, Technical Report NDN-0001, 2010.
- [6] 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*, December 2009, pp. 1–12.
- [7] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, “MobilityFirst: A robust and trustworthy mobility-centric architecture for the future internet,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 3, pp. 2–13, December 2012.

- [8] 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.
- [9] D. Kulinski and J. Burke, “NDNVideo: Random-access live and pre-recorded streaming using NDN,” NDN, Technical Report NDN-0007, September 2012. [Online]. Available: <http://www.named-data.net/techreport/TR007-streaming.pdf>
- [10] B. Han, X. Wang, N. Choi, T. T. Kwon, and Y. Choi, “AMVS-NDN: Adaptive Mobile Video Streaming with Offloading and Sharing in Wireless Named Data Networking,” in *Proceedings of IEEE INFOCOM NOMEN workshop 2013*, April 2013.
- [11] T. Braun, V. Hilt, M. Hofmann, I. Rimac, M. Steiner, and M. Varvello, “Service-centric networking,” in *Proceedings of IEEE International Conference on Communications Workshops 2011*, June 2011, pp. 1–6.
- [12] B. Saadallah, A. Lahmadi, and O. Festor, “CCNx for Contiki: implementation details,” INRIA, Technical Report RT-0432, November 2012. [Online]. Available: <http://hal.inria.fr/hal-00755482>
- [13] Y. Song, H. Ma, and L. Liu, “Content-centric internetworking for resource-constrained devices in the Internet of Things,” in *Proceedings of IEEE International Conference on Communications 2013*, June 2013, pp. 1742–1747.
- [14] “CCNxCon,” PARC, 2013. [Online]. Available: <http://www.ccnx.org/events/>
- [15] “CCNx,” PARC, 2013. [Online]. Available: <http://www.ccnx.org/>