

マイクロコントローラ向け NDN プロトコルスタックの設計と実装

加藤 颯^{*} 松澤 智史^{*}

東京理科大学 創域理工学部 情報計算科学科^{*}

1 研究背景

現在のインターネットは IP (Internet Protocol) を基盤とするホスト中心のアーキテクチャで構築されている。しかし、主要な利用形態は特定のホストとの通信から、動画配信や SNS といったコンテンツの取得へと変化している (図 1)。Information-Centric Networking (ICN) は、この変化に対応するためネットワークの根本をコンテンツ中心に再設計する研究分野である [1]。ICN では、IP アドレスの代わりにコンテンツの名前を用いてデータを要求し、ネットワーク内キャッシングによりルータがデータを保持して効率的に応答できる。

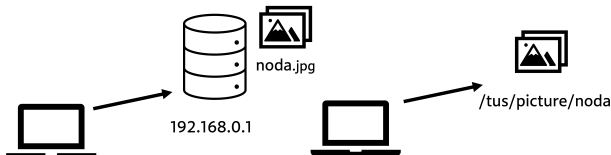


図 1: IP によるホスト指定と ICN による名前指定

一方、地震や水害などの災害時には既存の通信インフラが利用できなくなる。Mobile Ad-hoc Network (MANET) [2] は基地局や固定インフラに依存せず、端末同士が直接通信しながらネットワークを構成する方式であり、災害時の通信手段として有効である。ICN は MANET と相性が良い。IP ネットワークではアドレス管理やルーティングの収束が必要だが、ICN では名前でデータを取得するためトポロジの変化に柔軟に対応できる。

IoT の普及により、ESP32 のような安価なマイクロコントローラがネットワークに接続される機会が増えている。こうしたデバイスで ICN を動作させれば、インフラに依存しないアドホックネットワークを低コストで構築できる。しかし、既存の ICN 実装は数百 MB のメモリを必要とする PC 向けのもが多く、マイクロコントローラでの動作は想定されていない。ICN の代表的なアーキテクチャに Named Data Networking (NDN) がある。NDN では名前によるデータ要求とネットワーク内キャッシングにより、コンテンツ指向の通信を実現する。本研究では、NDN プロトコルスタックをマイクロコントローラ向けに設計および実装し、安価なデバイスのみで MANET 上の NDN 通信を可能とする。

2 基礎知識

2.1 Named Data Networking (NDN)

NDN [3] では、Consumer が Interest (要求パケット) を送信し、それに対して Producer が Data (データパケット) を返す形で通信する (図 2)。

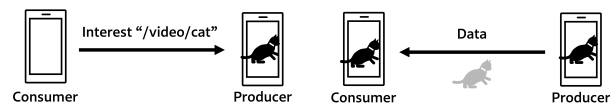


図 2: NDN の通信モデル

NDN ノードは 3 つのデータ構造を持つ。Pending Interest Table (PIT) は「誰がどのデータを待っているか」を記録する。Forwarding Information Base (FIB) は「どの名前はどの方向に転送すればよいか」を記録する。Content Store (CS) は通過した Data をキャッシュし、同じ Interest が来たらネットワーク内から直接応答できる。

2.2 State Vector Sync (SVS)

State Vector Sync (SVS) ¹ は、NDN 上で動作するデータ同期プロトコルである。各ノードは自身の最新シーケンス番号を状態ベクトルとして保持し、これを交換することで新しいデータの発行を検知および取得する。

3 関連研究

NDN Forwarding Daemon (NFD) ² は、NDN プロジェクトが開発するリファレンス実装である。PIT、FIB、CS を完備し複数の転送戦略を提供するが、汎用 OS 上での動作を前提としており、数百 MB のメモリが必要である。

esp8266ndn³ は、Arduino 向けの NDN ライブラリである。ESP8266 や ESP32 上で動作するが、PIT、FIB、CS を持たないテーブルレス設計のため、マルチホップ転送には対応していない。

本研究は、PIT、FIB、CS を備えた完全な Forwarder をマイクロコントローラ上に実装する点で esp8266ndn と異なる。また、Arduino ではなく ESP-IDF を使用し、FreeRTOS や ESP-NOW⁴ の低レベル API を活用する。

4 提案手法

全体構成を図 3 に示す。本プロトコルスタックは次のように設計する。

- 動的メモリ確保を避け静的バッファを使用し、限られたメモリで安定動作させる

¹<https://named-data.github.io/StateVectorSync/Specification.html> 閲覧日: 2025/12/19

²<https://github.com/named-data/NFD> 閲覧日: 2025/12/19

³<https://github.com/yoursunny/esp8266ndn> 閲覧日: 2025/12/19

⁴<https://github.com/espressif/esp-now> 閲覧日: 2025/12/19

Design and Implementation of an NDN Protocol Stack for Micro-controllers

^{*}Kato So, Matsuzawa Tomofumi, Tokyo University of Science

2. 各コンポーネントをモジュール化し依存関係を明確にする
3. 通信方式をFaceとして抽象化し、ESP-NOW 以外にも対応可能とする

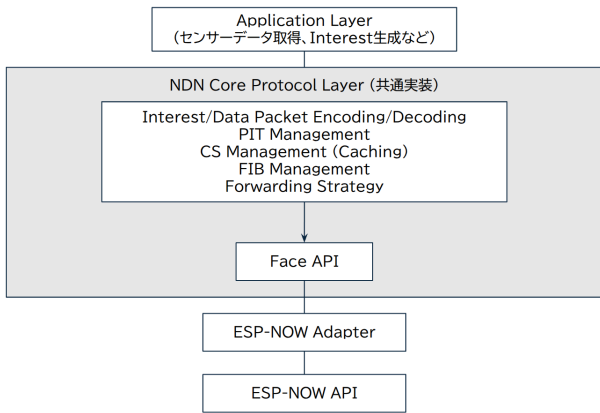


図 3: プロトコルスタックの全体構成

主要コンポーネントとして、PIT (最大 50 エントリ)、CS (最大 20 エントリ、LRU 置換)、FIB (最大 30 エントリ) を実装した。ESP-NOW Face はパケット受信時に送信元 MAC アドレスを自動でピア登録する。転送戦略はブロードキャスト方式を採用し、明示的なルーティングプロトコルなしでマルチホップ転送を実現する。

5 実験と結果

デバイスには M5Stack AtomS3 Lite (ESP32-S3FN8 搭載) を使用し、通信方式には ESP-NOW を採用した。開発環境は ESP-IDF v5.5 である。

5.1 物理的マルチホップ転送

物理的に電波が届かない距離 (約 50 m) に Consumer と Producer を配置し、中間に Router を投入することでマルチホップ転送が成立することを検証した。図 4 に Router 投入前後の通信結果を示す。投入前は成功率 0% だったが、投入後は 76.9% (13 回中 10 回成功、平均 RTT 61 ms) となり、撤去後は再び 0% に戻った。成功率が 100% でなかったのは長距離通信による電波品質の低下が原因であり、至近距離での補足実験では 100% を達成した。

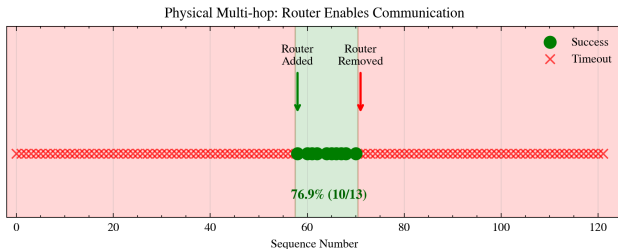


図 4: マルチホップ転送実験の結果

5.2 キャッシュ効果の測定

Consumer、Router、Producer の 3 台を直列に配置し、CS のキャッシュ効果を測定した。同一 Interest を 3 回連続送信し、1 回目 (キャッシュミス) と 2 回目以降 (キャッシュヒット) の RTT を比較した (図 5)。キャッシュミス時は平均 54 ms、キャッシュヒット時は 28 ms で、約 48% の RTT 削減

を確認した。この差分 26 ms は Producer までの 1 ホップ往復に相当する。

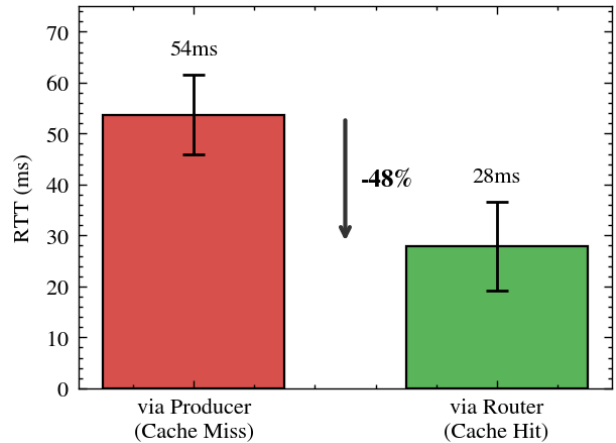


図 5: キャッシュ効果の測定結果

5.3 SVS v3 によるメッセージ同期

本プロトコルスタック上でアプリケーションが動作することを示すため、SVS v3⁵ を実装し評価した。5 台のデバイスを ESP-NOW ブロードキャストによるフルメッシュ構成とし、各ノードが 5 秒間隔でメッセージを送信する 300 秒間の連続動作を行った。平均メッセージ到達率は 99.5%、同期遅延の中央値は 58 ms であった。パケットロスが起きやすい ESP-NOW ブロードキャスト環境でも、SVS の再送機構により高い信頼性を達成した。

6 まとめ

本研究では、マイクロコントローラ向け NDN プロトコルスタックを設計および実装し、アドホックネットワーク上での動作を検証した。提案手法は ESP-IDF 上でネイティブに動作し、PIT、FIB、CS を備えた完全な Forwarder 機能を提供する。静的バッファの使用により、ESP32-S3 の限られたリソースで安定動作を実現した。

実験では、マルチホップ転送の成立、キャッシュによる RTT 約 48% 削減、SVS v3 による到達率 99.5% のメッセージ同期を確認した。これらの結果から、マイクロコントローラのみで NDN 通信と分散データ同期が実現可能であることを示した。

参考文献

- [1] G. Xylomenos et al., "A Survey of Information-Centric Networking Research," IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 1024-1049, 2014.
- [2] S. M. Corson and J. P. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, 1999.
- [3] L. Zhang et al., "Named data networking," SIGCOMM Comput. Commun. Rev., vol. 44, no. 3, pp. 66-73, 2014.

⁵本稿執筆時点での最新仕様である。