組み込みエッジ AI 向け TCP/IP プロトコルスタックの研究

笹崎　海利[1]　並木　美太郎[1]

1. はじめに

近年，低価格かつ小型の省電力デバイスで人工知能（AI）を動作させる組み込みエッジ AI が登場している． エッジ AI のような IoT デバイスを使ってデータを収集するための通信プロトコルに MQTT がある．MQTT はヘッダサイズが小さくパブリッシュ/サブスクライブ・メッセージングモデルのプロトコルとして注目されている．

しかし，MQTT で大量のエッジ AI をセンサノードとして用いてデータ収集を行う際に，一部のエッジ AI デバイスは AI としての動作は十分可能だがネットワークに接続することができないという場合がある．これは，デバイスがネットワークに接続するためには TCP/IP プロトコルスタックの移植に加え，ネットワークインタフェース層で利用するプロトコルを実装する必要があるからである．だが，RAM や ROM，計算資源などにおいて限られたリソースしか通信に使用できないエッジ AI や，もとより省リソースの小型デバイスに対して一般にオペレーティングシステム（OS）で備えられているような高度な通信機能を持ったプロトコルスタックを実装することは困難である．

これに対し，本研究では組み込みエッジ AI に軽量 TCP/IP プロトコルスタックである uIP[1] の移植とネットワークインタフェースの実装をすることでネットワークへ接続し，管理システムとの間において TCP/IP 通信を可能にすることを目的とする．

2. 関連研究と提案手法

小型デバイスと軽量 TCP/IP プロトコルスタックを利用し，センサネットワークを実現した研究 [2] がある．この研究ではネットワークインタフェース層のプロトコルに Ethernet が使用されている．Ethernet を利用する際，フレームサイズはデフォルトで 1.5K バイトに設定されているが，小型デバイスでこの Ethernet を使用する場合次に示す課題がある．

- 表 1 に示すように最小 2K バイトしかない小型デバイスの RAM を Ethernet フレームが圧迫してしまう．
- Ethernet コントローラなどのハードウェアがない．
- 通信に利用可能な計算資源が乏しい．

これらのことから，Ethernet を必要とせずに TCP/IP プロトコルスタックを利用可能にすることが必要である．そこで，小型デバイスでも多くがシリアルインタフェースを持っていることに着目し，これを Ethernet の代用とした．シリアル通信を用いて 2K バイトの RAM 容量しかない Arduino で uIP を動作させている例 [3] もあり，RAM 使用量の削減が期待出来る．

本研究では，物理層に UART，そしてデータリンク層に SLIP を用いることで TCP/IP 通信を扱う．さらに，OS を必要としないベアメタルの移植が可能な uIP を利用して組み込みエッジ AI のネットワーク接続を実現する．

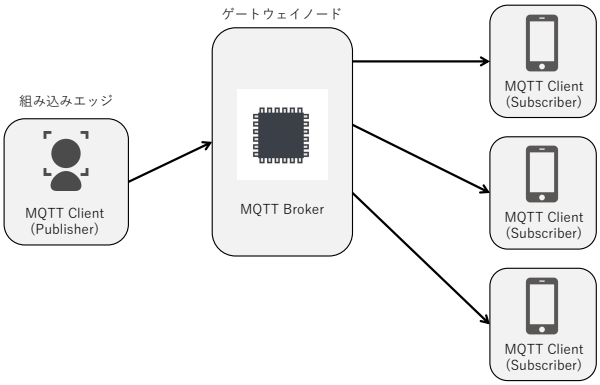

図 1　MQTT を利用した IoT システムの構成図

3. 全体構成

MQTT の利点は 1 対多数の通信が可能であることにある．シリアルラインで Point-to-Point（P2P）接続をした場合，本来データを取得できるのは接続先のみである．ただし，**図 1** に示すように組み込みエッジ AI をゲートウェ

[1] 東京農工大学
Tokyo University of Agriculture and Technology

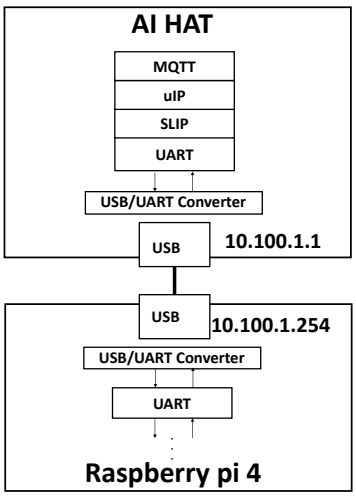

図 2 P2P 接続のプロトコル構成図

イに接続し，このゲートウェイを MQTT ブローカとすることで任意のサブスクライバにデータを提供することが出来るようになる．

現在までに AI アクセラレータを搭載し，RISC-V ベースの SoC である k210 を搭載した開発ボードの Grove - AI HAT for Edge Computing（以降 AI HAT と呼ぶ．）に対して uIP の移植を行っている．この AI HAT をパブリッシャ，AI HAT のゲートウェイである Raspberry pi 4 をブローカとして MQTT を利用した IoT システムを構成する．

4. 設計

本研究では**図 2** のプロトコル構成図に示すように物理層に UART，データリンク層に SLIP を利用することに加え，さらに上位の層では最低 10K バイトのコード，数百バイト程度のメモリ使用量で動作可能なオープンソースの軽量 TCP/IP プロトコルスタックである uIP を利用することで TCP/IP 通信を実現する．そして，移植した uIP 利用して IoT システム向けアプリケーションプロトコルである MQTT を動作可能にする．

AI HAT と Raspberry pi 4 は I/O ピンを使用して直接 UART 接続することが可能だが現在はどちらも USB で接続しており，ボードに組み込まれた変換器で USB と UART が変換されている．接続された AI HAT と Raspberry pi 4 のシリアルラインにおいて slattach コマンドにより SLIP を有効化する．AI HAT の IP アドレスと，Raspberry pi 4 の IP アドレスをそれぞれ設定することでシリアルライン上で P2P の IP ネットワークを実現する．

5. 実装

今回利用する uIP はマイクロコントローラで TCP/IP 通信を行うために開発された BSD ライセンスのオープンソースプロトコルスタックである．uIP はメインループの中で TCP/IP を動作させ，メインループ内ではパケットの確認とタイムアウトの確認を行っている．OS なしのベアメタルで動作するため，uIP とハードウェアの間にはネットワークインタフェースの実装が必要となる．

ネットワークインタフェース部分ではシリアルデバイスの I/O を制御すると同時に SLIP を利用できるようデータをフレーム化する．シリアル通信方式の中でも今回は UART を利用している．そのため，k210 の SDK に用意されている UART 制御 API の read（）,write（）を利用して実装する．データのフレーム化は SLIP では特殊文字の END でフレームを終えることで行う．

表 1 実行時性能の評価

	速度 (KBps)	RAM 使用量 /RAM 容量 （K バイト）	コードサイズ /ROM 容量 （K バイト）
AI HAT	1.0	$10/6.0\times10^3$	$14/16\times10^3$
Arduino	4.1	1.2/2.0	7.3/32

6. 評価

表 1 は AI HAT と Arduino に uIP を移植し，telnet サーバとして動作させたものの性能評価を示したものである．RAM 容量の小さな小型デバイスで OS を動作させ TCP/IP 通信を実現することは困難であるが，表 1 の RAM 使用量を見ると提案手法により TCP/IP 通信を実現することは十分可能であると分かる．また，パケットサイズを変えながら送信時間を計測しており，今回ボーレートは 115200 としているため転送速度を 115200bps とすれば，最高 14.4KBps のシリアルラインである．ただし，結果を見ると帯域の限界に到達していないということが分かった．そのため，転送速度低下の原因となる何らかのオーバーヘッドがあり，原因の一部は Arduino と比較しても AI HAT の転送速度が低いことから k210 の UART 制御 API を使用している点にあると考えられる．

7. おわりに

本稿では，組み込みエッジデバイスにおけるシリアルインタフェースを利用した TCP/IP 通信の実現について述べた．今後はアプリケーションプロトコルである MQTT の実装に着手し，IoT システムを実現していく．

参考文献

[1] A Dunkels, “The uIP embedded TCP/IP stack”, Swedish Institute of Computer Science, 2006
[2] Devika.S, Rajesh.S, Ananda.S,“Prototype of IoT Implementation Based On LwIP Stack Protocol & SWE Standard”, International Journal of Engineering Development and Research, 2015 Vol3 Issue1, PP414-421
[3] https://github.com/Inokinoki/SerialIP(accessed: Nov. 10, 2022)