

センサネットワークにおける通信モジュール再利用のための クロスレイヤ処理機構

金丸 達雄[†] 松尾 英治^{††} 鈴木 和久^{††} 横田 裕介[†] 大久保 英嗣[†]

[†]立命館大学情報理工学部 ^{††}立命館大学大学院理工学研究科

1 はじめに

無線センサネットワークでは、センシング結果を基地局に通知する時間の短縮、ノード間の通信における信頼性確保、符号化を行うハードウェアの利用といった要求に応じた様々なプログラムを開発する必要がある。このようなプログラムをコストを抑え、短期間で開発するためには、既存のモジュールを可能な限り再利用し作業量を抑えることが重要となる。

また、センサネットワークでは、有線による電力供給がないため、乾電池などの電力資源を効率よく利用する必要がある。そのため、ネットワークの状態に応じ通信の挙動を変更して転送効率を高め、無線通信時間を削減する手法が必要である。この手法の1つとして、クロスレイヤデザイン [1] による通信プロトコルが提案されており、本稿ではこれをクロスレイヤプロトコルと呼ぶ。クロスレイヤプロトコルは、太陽電池や ASIC といった電力資源の改善を行うハードウェアを必要としない。そのため、ノードに用いるハードウェアのコストを抑えることができる。これは、センサネットワークのような多数のノードを利用する環境において重要であると考えられる。

しかし、クロスレイヤプロトコルは、省電力化を重視するため、レイヤ間の独立性が低くなっている。そのため、汎用性が低く、再利用が困難である。このことにより、クロスレイヤプロトコルのモジュールの再利用は、開発期間の長期化につながり、開発コストを増加させると考えられる。

そこで、本研究ではクロスレイヤプロトコルのモジュールの再利用性を向上させる手法として、汎用的なクロスレイヤ処理機構を実現する。特に、本稿では、クロスレイヤ処理機構のソフトウェア構成について述べる。

2 クロスレイヤデザインにおける課題

クロスレイヤデザインによるプロトコルモジュールは、クロスレイヤ処理部においてレイヤバイオレーションが発生しており、また、実装毎にクロスレイヤ処理部を持つ機能やインタフェースが異なる (図 1 参照)。そのため、汎用性が低く再利用が困難である。

一方、センサネットワークでは、様々な要求に応じた開発を行う必要がある。この場合、クロスレイヤプロトコルのモジュールを汎用化し、再利用を行うことが重要である。このため、従来、開発者毎に異なるクロスレイヤ処理部を統一する必要がある。また、プロトコルスタックの階層間の独立性を保つため、クロスレイヤ処理部をプロトコルスタックの外部から提供する必要がある。プロトコルスタックに提供するクロスレイヤ処理部は、クロ

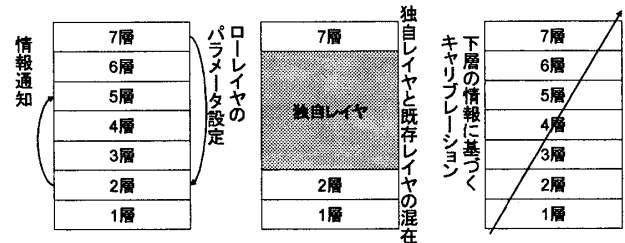


図 1 クロスレイヤ実装の例 (文献 [1] より抜粋)

スレイヤプロトコルからクロスレイヤ処理部を切り出したものである。そのため、次の機能を持つ必要があると考えられる。

- フィードバック機構
通信履歴からネットワークの状態を判別する。
- フィードバック結果の伝播管理
フィードバック結果を必要な層に通知する。
- 通信モジュールの制御
通知されたフィードバック結果に対するモジュールの制御を行う。使用する機能の切替え、通信の協調を目的とした隣接ノードに対するモジュールの状態通知を行う。

このような機能を持つクロスレイヤ処理部をプロトコルスタックが統一して持つことで、独自形式のクロスレイヤ処理部を前提としたモジュールを使用する必要がなくなる。そのため、クロスレイヤプロトコルを汎用的なモジュールとして構成することができると考えられる。

3 クロスレイヤ処理機構

3.1 クロスレイヤ処理機構の設計

提案するクロスレイヤ処理機構は、フィードバック機構、フィードバック結果に基づく動作を記述したルール、モジュール制御機能を持つプロトコルスタック管理層から構成される。本機構は、図 2 のようにプロトコルスタックの外部において構成される。

フィードバック機構は、通信履歴を基に特性 (トラフィックの集中、送受信時間の偏り、再送の頻発といった識別子) を検出する。また、検出した特性をプロトコルスタック管理層に通知する。

ルールは、フィードバック機構から通知される特性に対し、プロトコルスタック管理層が行う動作を記述する。ルールで指定される主な項目を表 1 に示す。これらの項目を用い、図 3 のように、自身または他ノードにて発生した特性に対して、対処する層、その層の送受信関数の切替え、対処するノード、特性の検出をトリガとして呼ばれる関数を指定する。

プロトコルスタック管理層は、特性とルールを照合し、プロトコルスタック内部の通信モジュールの切替え、ルールに示されている特性の検出をトリガにした関数の呼出し、隣接ノードへの特性の通知を行う。フィードバック機構から特性が通知された場合、ルールと特性を照合す

A Cross-Layer Processing Architecture for Reusable Communication Modules in Sensor Networks

Tatsuo Kanamaru[†], Hideharu Matsuo^{††}, Kazuhisa Suzuki^{††}, Yusuke Yokota[†] and Eiji Okubo[†]

[†]College of Information Science and Engineering, Ritsumeikan Univ.

^{††}Graduate School of Science and Engineering, Ritsumeikan Univ.

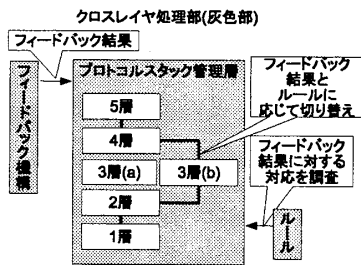


図2 提案機構の概要

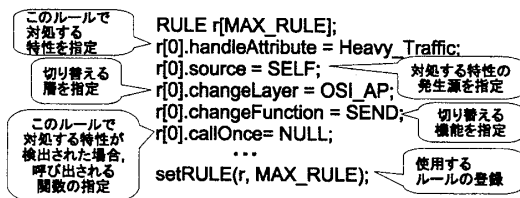


図3 ルールの指定例

る。次に、特性に対処するルールに記述された関数を呼ぶ。その後、プロトコスタックの各層のモジュールを切り替え、隣接ノードへ特性を通知する。隣接ノードの管理層は同様に、通知された特性とルールを照合することにより、送信元のノードのプロトコスタックの各層の状態を判別し、通信に使用するモジュールを切り替える。

これらの機構が連携することにより、フィードバック機構は、各層へのフィードバック情報の通知を実現し、プロトコスタックに対して統一したフィードバックを実現する。また、ルール記述によって、特性とプロトコスタック管理層の動作を対応付けることで、フィードバック先の管理が可能となる。さらに、プロトコスタック管理層は、ルールと特性の照合により、プロトコスタックのモジュールの統一した制御を実現する。

3.2 クロスレイヤ処理機構の適用例

子ノードからのトラフィックが集中しているノードは、親ノードに転送するデータ量の増加が考えられる。この場合、子ノードからのデータ受信後、即座に転送するのではなく、複数のデータを保持し、圧縮後転送することにより送信で消費する電力を削減できる。提案機構を用いこの手法を実現する場合を考える。処理手順は、以下のようなになる(図4参照)。

1. ノード A のフィードバック機構は、プロトコスタック管理層が通信毎に取得した履歴を調査し、トラフィックが集中しているか判定する。トラフィックの集中を検知した場合、プロトコスタック管理

表1 ルールで指定する項目の例

特性	このルールで対応するネットワークで発生している事象。トラフィックの集中, 送受信時間の偏り, 再送の頻発など。
特性発生元	このルールがどのノードからの特性の通知に対応するかを示す。隣接ノード, 自身, 両方, 任意のノード。
変更対象層	使用しているプロトコスタックの何層目を切り替えるか。
トリガ関数	特性検出時に呼び出される関数の指定。

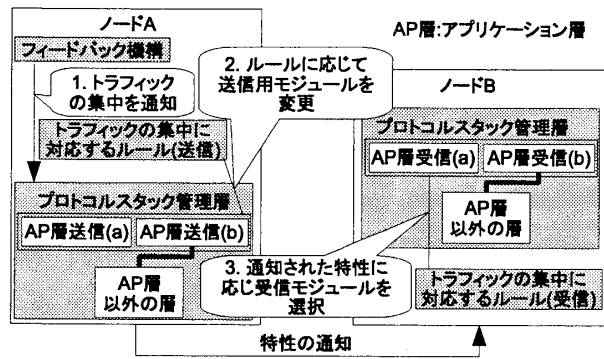


図4 トラフィックの集中に対応する例

層にこの特性を通知する。

2. A の管理層は、トラフィックの集中に対処するルールに従い、アプリケーション層の送信モジュールを変更する。これにより、A は、データ送信前に圧縮を行う。
3. A は、隣接ノード B に特性を通知する。特性を受信した B は、ルールと A から通知された特性を照合し、A から受信を行う際にアプリケーション層の受信モジュールを切り替える。これにより、B は、A からのパケットを受信した際に復号を行うことが可能となる。

4 クロスレイヤ処理機構の実装

実装は、TinyOS [2] 上で行う。無線通信を抽象化したコンポーネント (RadioCRCPacket) の上に提案機構のコンポーネントを作成する。

フィードバック機構と通信履歴の取得を FeedBackM コンポーネントで実現する。FeedBackM は、FeedBack インタフェースにより外部に履歴の更新や特性の検出通知といった機能を提供する。なお、ノード毎に通信履歴を取得するため、TOS_Msg 内のノードのアドレスが格納されている変数を用い、ノードの識別を実現する。

さらに、通信モジュールの制御、ルールの受理を ProtocolManagementM で実現する。ProtocolManagementM は、PManagement インタフェースにより、アプリケーションに対して送受信、ルールの登録を行う機能を提供する。ルールの構造の定義は、CrossLayer.h で行う。また、ProtocolManagementM は、ProtocolStack インタフェースを持ち、接続されたプロトコスタックのモジュールを用いて通信を行う。

なお、提案機構では、すべてのノードで同じルールと通信モジュールを保持する必要がある。これは、モジュールの挙動変更によって通信プロトコルが変更になった際に、ノード間で通信不能になることを回避するためである。

5 おわりに

本稿では、センサネットワークにおける通信モジュール再利用のためのクロスレイヤ処理機構について述べた。今後の課題として、TinyOS 以外の OS への移植を行う。

参考文献
[1] Srivastava, V., Motani, M., Cross-layer design: a survey and the road ahead, Communications Magazine, IEEE, 2005 Vol.43, pp.112-119, 2005.

[2] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister, System Architecture Directions for Networked Sensors, ACM SIGPLAN Notices, Volume 35, pp. 93-104, Nov. 2000.