

センサーネットワークの設計開発を支援する シミュレーション融合型テストベットの検討

池田 和史† 大田 雄也‡ 森 駿介‡ 梅津 高朗† 山口 弘純† 東野 輝夫†

† 大阪大学 大学院情報科学研究科

‡ 大阪大学 基礎工学部情報科学科

{ k-ikededa, umedu, h-yamagu, higashino }@ist.osaka-u.ac.jp

{ yuy-ohta, s-mori }@ics.es.osaka-u.ac.jp

1 はじめに

ワイヤレスセンサーネットワーク (WSN) は気象観測や動物の生態調査など様々な用途に利用できる技術として注目されており、対象アプリケーションによって多様なプロトコルが提案されている。WSN プロトコルの設計、開発においてはプロトコルの実装に多くの労力を要するとともに、実環境での実証実験が極めて重要になる。そこで本研究では WSN プロトコルの設計開発、性能評価を支援するためのテストベットを提案し、その設計開発について述べる。提案テストベットは多くの WSN プロトコルの実装に利用できる実装支援 API や実環境と連携したシミュレータによる実験支援機能、無線通信端末 MOTE を利用した実環境デバッグと性能評価実験支援機能を提供することで、WSN プロトコルおよびアプリケーションの開発と性能評価に必要な時間と労力を大幅に削減することを目標とする。

本稿では提案テストベットの全体設計や実装支援 API について述べ、既存の WSN プロトコルである GPSR[1] の実装例をもとに実装支援機能の有用性について検討する。また、シミュレータ連携機能、実環境デバッグ機能についても言及する。

2 関連研究

MOTE 端末を利用した既存のテストベットには MoteLab [2] があるが、この研究では MOTE 端末へのプログラムの配布やインターネットを介した端末の共有の実現を目標としている。これに対し、我々の提案するテストベットでは WSN プロトコルの実装、シミュレーションによる性能評価、実環境での実験やデバッグといった、プロトコルの設計から開発までを総合的に支援することを目指している。WSN プロトコルの実装を支援するミドルウェアとして MATE[3] が提案されているが、ここで提供されている API はセンサーのデータを読み込む動作 (sense) やスタックにデータを載せる動作 (pushc)、パケットを送信する動作 (send) といった、非常に汎用的なものに留まっている。我々のテストベットでは既存の WSN プロトコルを詳しく分析、分類し、それぞれに共通の動作を抽出することで、プロトコルの特徴に応じた複雑な動作の実装を支援する API も提供することが可能である。例えば、GPSR などの位置情報を利用してルーティングを行う多くのプロトコルでは、自ノードよりも目的ノードに近い位置にあるノードにパケットを送信するという動作が共通に利用される。この動作は指定ノードの位置情報取得、指定 2 ノード間の距離計算、指定ノードへのパケット送信など複数の処理からなる複雑なものであるが、提案テストベットでは 1 つの API として提供されている。また、シミュレータと

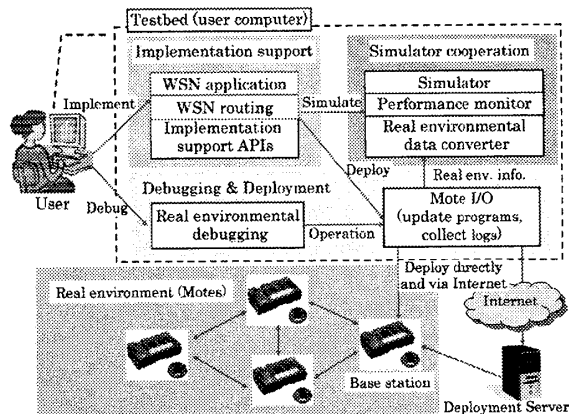


図 1: 提案テストベットの全体像

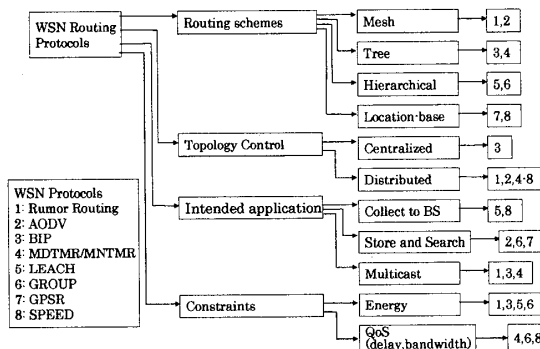


図 2: 既存 WSN プロトコルの分類

連携し、実環境と仮想環境をシームレスに繋ぐ開発環境を提供する点でも提案テストベットは大きく異なる。

3 テストベットの設計

テストベットの全体像を図 1 に示す。ユーザは実装支援 API を利用し、WSN プロトコルを容易に実装することができ、それらをシミュレーションにより評価したり、アドホックネットワークを介して MOTE 端末にプログラムを配布して、実環境におけるデバッグや性能評価も行うことができる。このように、提案テストベットは WSN プロトコルの実装から性能評価に至るまでの全工程を支援できるように設計されている。

3.1 実装支援機能

既存 WSN プロトコルはルーティング手法やトポロジの制御方式 (集中制御, 分散制御), 対象とするアプリケーション (データの基地局収集, 蓄積検索, マルチキャストによる配布), 考慮すべき制約などによって図 2 のように分類することができる。例えば、GPSR (図 2 の 7) はノードの位置情報を利用したルーティングを行い、データの蓄積と検索を行うアプリケーションに利用されるため、それぞれに特徴的な支援機能を組み合わせて実装できる。

Design of a hybrid testbed for development of wireless sensor networks
†Kazushi Ikeda ‡Yuya Ota ‡Shunsuke Mori †Takaaki Umedu
†Hirozumi Yamaguchi †Teruo Higashino
†Graduate School of Information Science and Technology, Osaka University
‡School of Engineer Science, Osaka University

汎用 API	
無線範囲内のノード ID を取得	get_neighbors()
隣接ノードにパケットを送信	send_packet(ID, pkt)
遠隔ノードまでのホップ数を取得	get_num_hops(ID)
Location base に特有の API	
指定ノードの座標を取得	get_location(ID)
指定 2 ノード間の距離を計算	get_distance(ID, ID)
指定 2 ノード間の角度を計算	get_angle(ID, ID)

表 1: プロトコルの実装支援 API 例

```

get_planar_graph(graph g){
  neighbor_IDs = get_neighbors()
  for all nodeID in neighbor_IDs
    for all nodeID' in neighbor_IDs
      if (get_distance(myID, nodeID)
          > max(get_distance(myID, nodeID'),
              get_distance(nodeID, nodeID'))))
        g.remove_edge(myID, nodeID)
}

```

図 3: RNG 作成アルゴリズム実装例

次に、WSN プロトコルに共通の支援機能を抽出した。支援機能は分類によらず利用される汎用的な機能と、各分類に特有の機能がある。表 1 は汎用 API と Location base のプロトコルに特有の API の例を示している。無線範囲内にあるノードの ID を取得する機能はどのプロトコルでも利用される汎用的な API であるのに対し、指定したノードの位置を取得する機能は Location base のプロトコルのみで利用される。このように、プロトコルを分類することで、より複雑な支援機能を提供することができる。

3.2 既存プロトコルの実装例

Location base のプロトコルで広く行われる平面的グラフ (planar graph) の作成やルーティングアルゴリズムの実装などの複雑な処理も表 1 に示される API を組み合わせることで容易に記述できる。図 3 に著名な平面的グラフである Relative Neighborhood Graph (RNG) 作成アルゴリズム [1] の API による記述例を示す。また、GPSR におけるルーティングを実装支援 API を利用して実装した例を図 4 に示す。このように、提案テストベットを利用することで、多くの WSN プロトコルで実行される動作の実装工程を省略することができる、直感的な記述が可能となる。

3.3 シミュレータ連携

WSN プロトコルの開発においてはシミュレーションによるプロトコル性能の見積もりや統計的データの取得が重要となる。実環境実験では実時間の実験が必要であるのに対し、シミュレーション機能を利用することで、実験にかかる時間を大幅に短縮することができ、統計的データが必要な評価も短時間で行うことができる。提案テストベットでは実環境で取得した情報、例えばノードの配置や電力残量といったノード状態、温度や気圧といった環境情報、センシングイベントなどをシミュレータで利用可能にする。これにより、ネットワークトポロジの構築やルーティング、ノード故障時の評価などにおいて、より現実性のあるシミュレーションが可能となる。

3.4 実環境デバッグ機能

提案テストベットはシミュレーションによるデバッグ機能だけでなく、実環境でのデバッグを支援する機能も提供する。電波強度の揺らぎや MOTE 端末の残存電力など、実環境に特有の要因がアプリケーションに及ぼす影響についても評価することができる。提供するデバッグ機能

```

// 目的ノード targetID により近いノードへパケットを転送
neighbor_IDs = get_neighbors()
for all nodeID in neighbor_IDs
  forwardID = myID
  if (get_distance(nodeID, targetID)
      < get_distance(forwardID, targetID))
    forwardID = nodeID
if (forwardID != myID) // より近いノード forwardID へ転送
  send_packet(forwardID, packet)
else // perimeter mode (省略)

```

図 4: GPSR 実装例

として、(1) ノード変数のモニタリングと変更、(2) ブレークポイントの設定とステップ実行などがある。

(1) ノード変数のモニタリングと変更では、ユーザがモニタしたいプログラム変数を明示することで、ユーザノード (センサネットワークに接続されたゲートウェイ PC) 上で、変数のモニタリングおよび値の変更を行うことができる。これにより、トポロジやノード状態を変更してのデバッグが可能となる。さらに、シミュレーションおよび実環境実験のログを読み込んで、実験のトレースを行うことも可能となる。

(2) ブレークポイントの設定とステップ実行では、ユーザがプログラム中にブレークポイントの条件を明記することで、集中ノードがシステムを停止させる。条件にはプログラム中の任意の場所を実行した場合、ノード変数が指定の値になった場合などがあり、これらを and, or 演算子により組み合わせる。例として、

```

(node1.x = 3) and (node2.y = 4) -> all.stop

```

と記述すると、ノード 1 の持つ変数 x が 3、かつ、ノード 2 の持つ変数 y が 4 になった場合、システム全体が停止する。また、記号 "\$" を用いて、条件に該当するノードの集合を表すこともできる。クラスタを持つプロトコルで、

```

($cluster_head = true) and
($num_cluster_member > 10) -> $.stop

```

のように記述すると、クラスタのメンバ数が 10 より多いクラスタヘッドを停止させることができ、ノード故障を想定したデバッグなども可能となる。

4 まとめと今後の課題

本稿では多くの WSN プロトコルの設計開発に利用できるシミュレーション融合型テストベットの設計についての検討を行った。提案テストベットでは既存の WSN プロトコルを詳しく分析することで、様々な種類の WSN プロトコルの実装を支援するための API を提供すると共に、シミュレーション実験や実環境におけるデバッグなど、プロトコル開発を総合的に支援する機能を提供する。今後は、提案したテストベットの実装を行い、既存プロトコルの性能をシミュレーションや実環境で比較評価することで、提案テストベットの有用性を示すことを考えている。

参考文献

- [1] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of MobiCom 2000*, pp. 149–160, 2000.
- [2] G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: a wireless sensor network testbed. In *Proc. of IPSN 2005*, pp. 483–488, 2005.
- [3] P. Levis and D. Culler. Mate: A tiny virtual machine for sensor networks. In *Proc. of ASPLOS-X2002*, pp. 85–95, 2002.