

Internet of Things を対象とした 大規模実証実験環境構築に関する研究

岩橋 紘司¹ 井上 朋哉² 篠田 陽一³

概要：インターネットは情報通信インフラとして定着しており、重要なサービスがインターネット上で提供されている。そのため、インターネットに接続するネットワークシステム一般にシステム展開前の十分な検証が重要である。

Internet of Things(IoT) の概念のもとで、多様なネットワークシステムがインターネットに接続されるようになりつつある。ワイヤレスセンサネットワークのような大規模分散配置を想定したネットワークシステムは、システムの展開後にハードウェアの回収を伴うシステムの更改を行うことが難しい。そのため、事前の検証が重要である。本研究では StarBED 型テストベッドにおいてハードウェアエミュレータを用いて実験環境を構築する際に必要となる、実験支援ソフトウェア群によるハードウェアエミュレータ制御手法と、ハードウェアエミュレータ間の通信およびその通信での無線通信の模倣を実現する手法 Generic Utilization of Assorted Networking (GUAN) を提案する。本論文では、IoT 指向のネットワーク実証実験環境構築に関する必要事項について考察し、GUAN の設計について議論を行う。

Research On Building An Internet-of-Things Oriented Large-Scale Experiment Environment

KOUJI IWAHASHI¹ TOMOYA INOUE² YOICHI SHINODA³

1. はじめに

インターネットは情報通信インフラとして社会に深く根ざしており、経済など社会的に重要なサービスがインターネットの上でも提供されている。そのため、新技術や製品の開発に際して十分な検証を行わず実際のインターネットに接続された場合、広く影響を与える問題が発生する可能性がある。新技術や新製品による悪影響の発生を未然に回避するために、ネットワークシステム一般の開発における事前の検証が重要である。

インターネットにおける検証は、新技術や製品がイン

ターネットに接続された際に正しく動作し他の技術や製品に悪影響を与えないかという動作確認と技術や製品の性能測定との2つが挙げられる。インターネットを利用する技術や製品の増加とともに、検証対象の技術や製品も増加する。

Internet of Things(IoT) [1] とは、一意に識別できるモノをインターネットに接続し、情報の収集やモノの制御を行う概念である。この概念のもとで、従来インターネットに接続されることの無かった製品に新しい利用法をもたらした一方で、他方では、事前に十分な検証をされずに製品が悪用される事態も発生している。IoT におけるモノとは、ネットワーク層にインターネットプロトコルを用いた通信機能を有するデバイスを広く指す。IoT においては、人間が介在せずに情報交換を行う自律的なデバイス同士の通信にもインターネットプロトコルが用いられる。それらのデバイスの中には、ハードウェアの消費電力に制約が存在し計算能力や通信能力が制限されたデバイスも存在する。IoT 指向のネットワークシステムの一例として、インターネットプロトコルを用いて通信を行うセンサを大規模分散

¹ 北陸先端科学技術大学院大学 情報科学研究科
School of Information Science,
Japan Advanced Institute of Science and Technology(JAIST)

² 北陸先端科学技術大学院大学
高信頼ネットワークイノベーションセンター
Dependable Network Innovation Center, JAIST

³ 北陸先端科学技術大学院大学
情報社会基盤研究センター
Research Center for Advanced Computing Infrastructure, JAIST

配置するワイヤレスセンサネットワークが挙げられる。このようなネットワークシステムではシステムの展開後にデバイスの回収を伴う更改をすることが困難であり、開発段階における検証が重要である。

StarBED [2, 3] は実際のインターネットに影響を与えずネットワーク技術の検証が可能な大規模ネットワーク実証実験環境である。本論文では StarBED と同様に PC とイーサネットを用いて構築された計算機クラスタと、実験支援ソフトウェア群で構成されたネットワーク実証実験環境を、StarBED 型テストベッドと呼ぶ。StarBED 型テストベッドは、実験者に実際のインターネットや他の実験者から独立したネットワークを提供する。SpringOS はネットワークシステムの大規模実証実験における環境構築および実験遂行に用いられる実験支援ソフトウェア群である。SpringOS を用いることでテストベッドの実験リソースに対する制御、実験環境の構築が容易になる。また、シナリオ駆動の実験遂行を行うことで、実験の再現性、可制御性を得ることが出来る。更に、施設のリソースが許容する範囲内で実験規模を設定することが可能である。

IoT 指向のネットワークシステムにおいては PC と異なるアーキテクチャで実装されたハードウェアが用いられ、通信手段についてもイーサネットや Wi-Fi 以外に、IEEE 802.15.4 [4] が用いられる傾向がある。つまり、IoT 指向のネットワークシステムには異種のハードウェア、異種の通信規格が混在する。このようなシステムの開発段階ではハードウェアエミュレータやネットワークシミュレータが用いられ、PC のリソースが許容する程度の規模でシステムの挙動を検証しつつ開発が進められる。しかし、大規模な展開を想定したシステムの動作確認や可用性の事前検証には、より柔軟に実験規模を設定可能な実証実験環境が求められる。そこで、我々は StarBED 型テストベッドにおいてハードウェアエミュレータを用いて実験環境を構築する際に必要となる、実験支援ソフトウェア群によるハードウェアエミュレータ制御手法と、ハードウェアエミュレータ間の通信とその通信での無線通信の模倣を実現する手法 Generic Utilization of Assorted Networking (GUAN) を提案する。本論文では、IoT 指向のネットワーク実証実験環境構築に関する必要事項について考察し、GUAN の設計について議論を行う。

2. StarBED 型テストベッドと IoT 関連技術

StarBED 型テストベッドはインターネットに影響を与えず、ネットワーク技術の検証が可能な大規模実証実験環境である。StarBED 型テストベッドにおいて用いられる実験支援ソフトウェア群 SpringOS は実証実験の環境構築と遂行を容易にする。

IoT 指向のネットワークシステムの開発においてハードウェアエミュレータやネットワークシミュレータは重要

な役割を担う。前者は、ハードウェアの調達をせずにアプリケーションの妥当性を検証できる。後者は、ハードウェアエミュレータと組み合わせて、ネットワークの挙動をシミュレートできる。

2.1 StarBED 型テストベッド

StarBED 型テストベッドは PC とイーサネットを用いて構築された計算機クラスタと実験支援ソフトウェア群で構成される大規模ネットワーク実証実験環境である。StarBED 型テストベッドでは、実験者に対してインターネットや他の実験者の利用するネットワークから独立した実験ネットワークを構築可能である。StarBED 型テストベッドを実験支援ソフトウェア群である SpringOS と合わせて利用することでシナリオ駆動の実験遂行が可能となり、実験の再現性、可制御性を得ることが出来る。また、施設のリソースが許す範囲であるが、実験規模を柔軟に設定することが可能である。StarBED 型テストベッドが有するネットワーク実証実験に関する特性は、ネットワークシステム一般の検証において有用である。すなわち、IoT を対象としたネットワーク実証実験の場合にもこの特性が十分に付与されれば IoT 指向ネットワークシステムの研究開発において有益である。

2.2 SpringOS

SpringOS は大規模実証実験環境構築および実験遂行に用いられる実験支援ソフトウェア群である。スクリプト実行による実験駆動システム Kuroyuri [5] などユーザの実験駆動や環境構築を支援する機能と、施設の電源、ネットワークなど実験環境の資源を管理する機能を合わせてテストベッドが有する計算機群の利用を容易にする。SpringOS で仮想マシンを扱うための拡張である SpringOS/VM [6] や汎用仮想ノードによる SpringOS の動的実験規模拡張 [7]、SpringOS で様々な多重化技術を用いるための拡張である SpringOS/M-ext [8] などの拡張がある。

Kuroyuri を用いた実験駆動は指令を行うマスターエージェントと、指令を受けて実験ノードの各種処理を行うスレーブエージェントによって行われる。マスターエージェントである Kuroyuri Master(kuma) は各実験ノードに配置されたスレーブエージェント Kuroyuri Slave(kusa) と通信し指令を行うことで、各実験ノード上の各種処理を行う。Kuroyuri への駆動内容の指示は K という実験記述言語を用いて記述されたスクリプトである。

Kuroyuri の実験駆動モデルの概形を図 1 で示す。図で示したモデルは、指令を送る構成要素 mgmt 上で動作する kuma と 3 台の実験ノード node 1, node 2, node 3 でそれぞれ動作する kusa が通信し、実験対象ソフトウェア target 1, target 2, target 3 による実験の遂行を表現している。kusa による実験ノードの各種処理は POSIX の API を用いて行

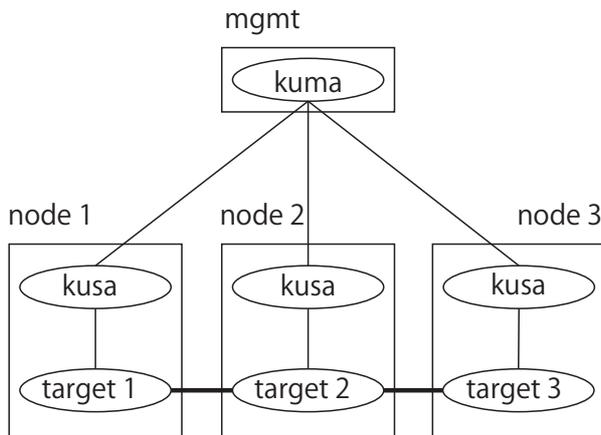


図1 Kuroyuriによる実験駆動モデルの概形

われる。そのため、POSIXのAPIが利用出来るノード一般においてKuroyuriが利用可能である。

2.3 ハードウェアの模倣

StarBED型テストベッドにおいて、PCと異なるアーキテクチャで実装されたIoT指向デバイスの実コードをそのまま実行することはできない。PC上で実コードを実行することの出来ない非PCのハードウェアを模倣する方法として、ハードウェアエミュレータを用いたエミュレーションがある。特に、IoT指向のネットワークシステムにおいては、技術とハードウェアの組み合わせが多岐にわたる。エミュレータは実際に利用されるアプリケーションのコードを実行することが出来るため、ハードウェアを調達することなくアプリケーションの妥当性を検証可能である。このため、ハードウェアエミュレータはIoT指向ネットワークシステムの開発においてしばしば利用される。実際に利用されるアプリケーションコードを実行出来るハードウェアエミュレータの例として、MSPSim [9], WSim [10], Avrora [11], TOSSIM [12] などがあ

2.4 無線通信の模倣

PCとイーサネットを用いて構築された実験環境で無線通信を利用するデバイスを模倣する場合、無線通信の模倣をする必要がある。無線通信による通信品質の変化を模倣する方法として、シミュレーションやエミュレーションがある。PC上でIoT指向無線ネットワークのシミュレーションを行うためのシミュレータはCOOJA [13]やWSNet [10]がある。COOJAはMSPSim, WSNetはWSimとの組み合わせで用いられ、無線ネットワークの状態をシミュレートする。ハードウェアエミュレータを用いて実際のアプリケーションで用いるコードによるネットワークの挙動が検証可能だが、ハードウェアエミュレータの種類を任意に選択することは出来ず、シミュレータで使用できるハードウェアのモデルが限定される。

3. StarBED型テストベッドにおけるIoT指向ネットワークシステムの実証実験

IoT指向のネットワークシステムを対象とした実験環境は、IoTを構成するデバイスや通信メディアの混在に対応し、デバイスの大規模分散配置に対応した環境であるべきである。しかしながら、実験に応じて大量のデバイスや大規模な空間を確保することは現実的に困難であり、PCを用いた模倣が行われる。この方法は一般的に、ハードウェアを模倣する機構と無線通信による通信品質の変化を模倣する機構とを組み合わせる。この方法の課題は、実験規模がPCのリソースに制約され柔軟に拡張できないことである。PCのリソースによる実験規模の制約に対して実験規模を拡張する方法は、PCのCPUやメモリなどの性能を向上させることで処理可能な実験規模を拡張する方法と、複数のPCを使用することで処理を分散させ、実験規模を拡張する方法がある。StarBED型テストベッドでの実験規模の柔軟な拡張のためには、複数のPCを使用する方法を選択すべきである。何故ならば、StarBED型テストベッドはPCとイーサネットで構築された計算機クラスタであり、多くのPCを予め有しているからである。

今後も多様なデバイスが出現することは想像に難くなく、IoT指向のネットワークシステムを構成するデバイスの数や種類が増加することが予想できる。すなわち、実証実験では多様なハードウェアエミュレータを用いて実験ノードの多様性が確保出来ることが望ましい。エミュレータの種類を問わず通信を実現することで、現実起きるIoT指向デバイスの増加に即して多様性のあるネットワークシステムを模倣することが容易になる。そのため、ハードウェアエミュレータの実装に関わらず使用可能であり、かつ、PCのリソースによる制約を受けず実験規模を柔軟に変更することが可能なIoT指向のエミュレーションフレームワークが必要である。

StarBED型テストベッドにおいて個々のハードウェアエミュレータを実験ノードとして扱うための課題は、ハードウェアエミュレータの制御とハードウェアエミュレータ間の通信にある。実験の開始や終了に際してハードウェアエミュレータを統一的に制御する必要がある。また、それぞれのエミュレータを実験ノードとして一意に識別し、同一のPC上で動作する複数のエミュレータ間や、異なるPC上で動作する複数のエミュレータ間での通信を実現する必要もある。

実験支援ソフトウェア群を用いて大規模実証実験を行うために、それぞれのハードウェアエミュレータが持つ制御されるべきインタフェースと実験遂行機構が管理ネットワークを通じて操作出来るインタフェースとの間で整合を図る必要がある。制御されるべきインタフェースとは、実験対象のハードウェアが有する物理的なボタンやセンサな

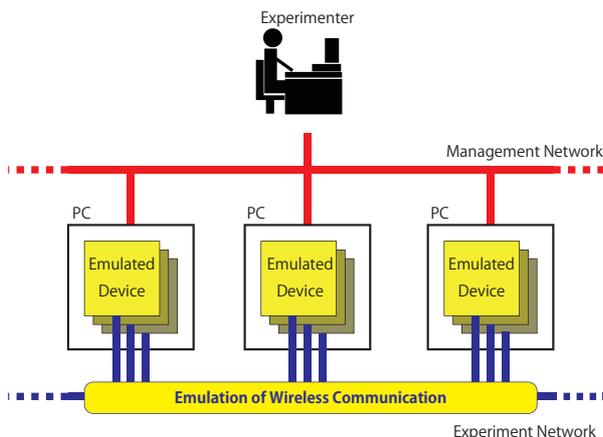


図2 IoT指向のネットワーク実証実験環境

ど、外部との物理的な相互作用に用いられるインタフェースを指す。制御されるべきインタフェースに対するハードウェアエミュレータの操作を抽象化し、実験遂行機構によるハードウェアエミュレータの統一的な制御を実現する。また、通信を実現する実験空間の定義が必要になる。これは、実験ノード間においてエミュレートする無線通信の成否の計算に必要である。制御のインタフェースと実験空間を定義することで、実証実験環境の構成は図2で示す形になる。実験に用いられるPCは、管理ネットワークと実験ネットワークのそれぞれに接続され、制御で用いられるパケットと実験ネットワークを流れるパケットとが混ざることはない。これにより、実験ネットワークを通じて行われる通信は、実験対象のネットワークシステムで観測される通信を模倣したものになる。

本研究では、任意のハードウェアエミュレータを複数用い、それらのエミュレータを統一的に制御する手法と、異なるPCの上で動作する任意のハードウェアエミュレータ間の通信を実現する手法、GUANを提案する。

4. Generic Utilization of Assorted Networking (GUAN)

図3はGUANのアーキテクチャの概観である。GUANは、機能毎に1)ハードウェアエミュレーションレイヤ、2)インタフェース調整レイヤ、3)配送機構レイヤ、の3つのレイヤに分けて構成される。本章ではこれら3つのレイヤの機能とハードウェアエミュレータが生成するL2フレームの取扱いについて述べる。

4.1 ハードウェアエミュレーションレイヤ

ハードウェアエミュレーションレイヤでは、操作性の異なるエミュレータを統一的に制御するためにインタフェースを抽象化する。

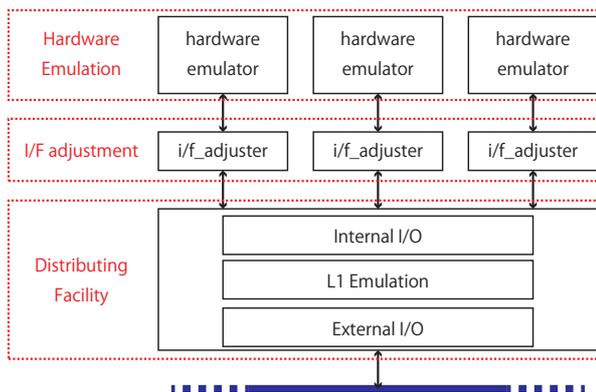


図3 GUANのアーキテクチャ

4.2 インタフェース調整レイヤ

インタフェース調整レイヤはエミュレータに対する入出力のインタフェース整合を実現する。多くの種類のエミュレータを一般的に利用可能にするため、エミュレータ毎に割り当てられる識別子やL2フレームの入出力方法など実装毎に異なる部分の抽象・汎化を実現する。異種のエミュレータ間における通信を実現するために、エミュレータの入出力を正規化し、GUANの配送機構で取り扱う。配送先のインタフェース調整機構によって入力対象であるエミュレータの形式へ変換する。

4.3 配送機構レイヤ

配送機構レイヤでは、ハードウェアエミュレータが生成するL2フレームの配送に係る一連の処理を行う。このレイヤは以下3つのサブレイヤで構成され、それぞれの機能を実現する。

1) 内部I/Oサブレイヤ

内部I/Oサブレイヤはエミュレータに対する制御命令やL2フレームの入出力を扱う。エミュレータから出力され、インタフェース調整レイヤの処理を受けたL2フレームはL1エミュレーションサブレイヤに渡される。また、L1エミュレーションサブレイヤから渡されたL2フレームは宛先のハードウェアエミュレータに向けて配送される。

2) L1エミュレーションサブレイヤ

L1エミュレーションサブレイヤは、GUANの構成するネットワーク上でノード間の通信に無線通信の模倣を適用する。StarBED型テストベッドにおける無線通信のエミュレーション手法としてQOMET [14]を利用する方法がある。QOMETはシナリオ記述に基づいて2点間の無線通信における通信品質の変質を計算し、リンクレベルのエミュレータを用いて有線のネットワーク環境において無線ネットワークの通信状態を模倣する。QOMETが無線通信の通信品質の変化を計算し生成する ΔQ というファイルを解釈し、GUANにおけるハードウェアエミュレータ間の通信において無線ネットワークの通信状態を模倣する。これに

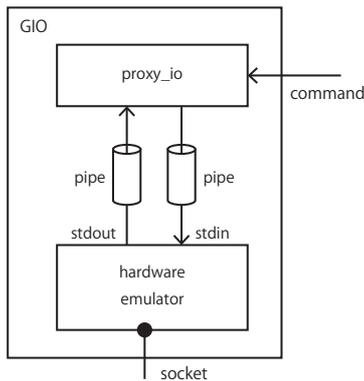


図 4 GIO の概形

よって、ハードウェアエミュレータ間の通信において、無線通信による通信品質の変化を模倣し、より現実に近い実験結果を得ることが可能となる。

3) 外部 I/O サブレイヤ

外部 I/O サブレイヤは、PC が接続されているネットワークに対する入出力を実現する。実験ネットワークにおいて、他の PC との間でハードウェアエミュレータが生成した L2 フレームの送受信を中継する。これによって異なる PC 上で動作するエミュレータ間の通信を実現する。

4.4 ハードウェアエミュレータが生成する L2 フレーム

無線通信を行うハードウェアのエミュレータは IEEE 802.11 や IEEE 802.15.4 のフレームを生成する。これらフレームの取り扱いにはエミュレータの実装によって異なる。そのため、異なるエミュレータ間での通信を実現可能にするために、L2 フレームの取り扱いを正規化する必要がある。また、有線イーサネットで作成される実験ネットワークに無線通信の L2 フレームをそのまま流すことは出来ない。そこで、GUAN においては、ハードウェアエミュレータが生成する L2 フレームをバイト列として取り扱い、これをカプセル化して配送する。

5. エミュレータ制御機構

ネットワークテストベッドにおける実験支援ソフトウェア群である SpringOS は、実験ノードとして物理マシンや仮想マシン、すなわちホストという形を想定している。つまり、物理マシンや仮想マシン上で実行されるプロセスであるハードウェアエミュレータを実験ノードとして細かく制御・管理することを想定していない。そのため、IoT 指向の実験を行うためのエミュレータを利用する場合、実験ノードとしてハードウェアエミュレータを制御する機構が必要になる。

図 4 は GUAN's I/O wrapper(GIO) の概形を示す。GIO は GUAN のアーキテクチャにおけるインタフェース調整レイヤの機能を実現する。エミュレータに極力加工を加えずに

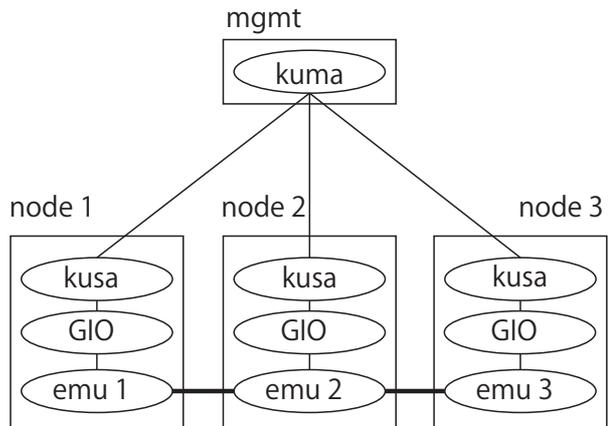


図 5 遂行機構に対応したハードウェアエミュレータの制御インタフェースの拡張

エミュレータの入出力を制御に利用するために、GIO はエミュレータのラッパーとして動作する。GIO はエミュレータを起動し、エミュレータの標準入出力をパイプする。パイプされた標準入出力は kusa によるエミュレータの制御に用いる。proxy_io は kusa によって中継された指令を受けてハードウェアエミュレータに対応した形式に変換したコマンドをハードウェアエミュレータに入力する。エミュレータがネットワークソケットを生成する場合は GIO で利用せず、別のインタフェース調整レイヤの機能でそれを利用する。GIO において、制御の対象とできるエミュレータは標準入出力でのコマンドラインによる制御を行うもの一般である。

図 1 で示した Kuroyuri による実験駆動モデルに、GIO を加えエミュレータの制御を行う拡張を図 5 で示す。図で示したモデルは、指令を送る構成要素 mgmt 上で動作する kuma と 3 台の実験ノード node 1, node 2, node 3 でそれぞれ動作する kusa が通信し、GIO を通じてエミュレータ emu 1, emu 2, emu 3 を制御する実験の遂行を表現している。GIO は kusa が直接行うことのできないエミュレータの制御を代行し統一的な制御を実現する。

6. 今後の展望

インタフェース調整レイヤの機能について、要件の定義と詳細な設計が必要である。また、配送機構レイヤにおける機能について、より詳細な要件定義と設計が必要である。特に、L1 エミュレーションの実現方法についてリンクエミュレータを用いず個々のハードウェアエミュレータプロセスに入力される L2 フレームに無線通信の模倣を適用する方法が求められる。加えて、IoT においてはセンサ類を始めとする実世界と物理的に相互作用するデバイスに入力する物理的な情報の模倣機構が求められる。

7. おわりに

本論文では、IoTを対象とした大規模実証実験環境構築に関する必要事項について考察し、実験規模が柔軟に変更可能なエミュレーションフレームワークの必要性を指摘した。StarBED型テストベッドを用いて柔軟な規模追従を実現するために、StarBED型テストベッドにおいてIoT指向ネットワークシステムを実験するための要件を考察した。

StarBED型テストベッドにおけるIoTを対象とした大規模実証実験環境構築のための手法GUANを提案しその設計に関する議論を行った。提案手法GUANは複数のPCで動作する複数のハードウェアエミュレータ間の通信を実現する。PC1台のリソースによる制約を受けずに実験規模が柔軟に拡張可能になる。GUANはStarBED型テストベッドと実験支援ソフトウェア群と合わせて用いることで、実験環境の設定やソフトウェアの展開、複数のハードウェアエミュレータに対する統一的な制御を実現する。

参考文献

- [1] Kevin Ashton. That ‘Internet of Things’ Thing. June 2009. 最初に Internet of Things という語が用いられたのは、RFID Journal の Kevin Ashton の記事である。
<http://www.rfidjournal.com/articles/view?4986>.
- [2] Toshiyuki Miyachi, Ken-ichi Chinen, and Yoichi Shinoda. Starbed and springos: Large-scale general purpose network testbed and supporting software. In *Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools*, valuertools '06, New York, NY, USA, 2006. ACM.
- [3] StarBED Project. <http://www.starbed.org/>.
- [4] IEEE Standard for Local and Metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks(LR-WPANs). *IEEE Std. 802.15.4-2011*, 2011.
- [5] 知念賢一, 宮地利幸, 篠田陽一. Kuroyuri: ネットワーク実験記述言語処理系. コンピュータソフトウェア, Vol. 27, No. 4, pp. 43–57, oct 2010.
- [6] 宮地利幸, 知念賢一, 篠田陽一. Springos/vm: 大規模ネットワークテストベッドにおける仮想機械運用技術(ネットワーク). 情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム], Vol. 48, pp. 105–112, 2005.
- [7] 知念賢一, 篠田陽一. 汎用仮想ノードによる SpringOS の動の実験規模拡張. 信学技報 IN2012-187, pp. 197–202, 2013.
- [8] 鍛冶祐希, 安田真悟, 知念賢一, 篠田陽一. Springos m-extension: 超大規模実証実験環境を想定したリソース指向実験ノード管理手法. *DICOMO 2012*, Vol. 48, pp. 1513–9519, 2012.
- [9] Joakim Eriksson, Adam Dunkels, Niclas Finne, Fredrik Osterlind, and Thiemo Voigt. Mspsim—an extensible simulator for msp430-equipped sensor boards. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*, 2007.
- [10] Guillaume Chelius, Antoine Fraboulet, and Eric Fleury. Worldsens: A fast and accurate development framework for sensor network applications. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pp. 222–226, New York, NY, USA, 2007. ACM.
- [11] Ben L Titzer, Daniel K Lee, and Jens Palsberg. Avroa: Scalable sensor network simulation with precise timing. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, p. 67. IEEE press, 2005.
- [12] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: Accurate and scalable simulation of entire tinyos applications. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 126–137. ACM, 2003.
- [13] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-Level Sensor Network Simulation with COOJA. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pp. 641–648. IEEE, 2006.
- [14] Razvan Beuran, Junya Nakata, Takashi Okada, Lan Tien Nguyen, Yasuo Tan and Yoichi Shinoda. “A Multi-purpose Wireless Network Emulator: QOMET”. In *Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on*, pp. 223–228, March 2008.