

軍隊ヒエラルキーに基づくスケーラブルなIoTセンシング基盤の検討

井元 滉[†] 梶本 真佑[†] 佐伯 幸郎[†] 中村 匡秀[†]

[†] 神戸大学大学院システム情報学研究科 〒657-8501 神戸市灘区六甲台町 1-1

E-mail: †inomoto@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp, †††sachio@carp.kobe-u.ac.jp

あらまし IoTにおけるセンシングを実現するためには、膨大な数のモノのセンシング結果をスケーラブルに管理し、かつこれらのモノを容易に管理可能な基盤が必要となる。本研究ではIoTセンシングを実現する方法として、軍隊のアナログを取り入れたセンシング基盤、センシング部隊を提案する。センシング部隊では、IoTにおける個々の「モノ」を「兵（二等兵）」と見立てる。さらに、兵を分隊単位で管理することでスケーラビリティを達成し、分隊内での命令の伝達を再帰的に行うことで設定の容易性を達成する。本稿ではセンシング部隊のコンセプトと具体的なアーキテクチャの提案、及び実現可能性のために試作したプロトタイプの紹介を行う。

キーワード センシング部隊, IoT, 環境センシング, 軍隊のヒエラルキー, スケーラビリティ, 設定容易性

A Study on Scalable IoT Sensing Platform based on Army Hierarchy

Hikaru INOMOTO[†], Shinsuke MATSUMOTO[†], Sachio SAIKI[†], and Masahide NAKAMURA[†]

[†] Kobe University, Rokkodai 1-1, Nada, Kobe, Hyogo, 657-8501 Japan

E-mail: †inomoto@ws.cs.kobe-u.ac.jp, ††{shinsuke,masa-n}@cs.kobe-u.ac.jp, †††sachio@carp.kobe-u.ac.jp

Abstract In order to realize sensing in an era of IoT, it is required that scalable management of environmental sensing data retrieved by extremely massive amount of things. Additionally, configurability of these things must be achieved. This paper proposes a uniformed IoT sensing framework, named “Sensing Troops”, which introduces a concept of army. In Sensing Troops, every “thing” is regarded as a “private soldier”. This framework achieves high scalability by managing private soldiers in unit of squad. Furthermore, high configurability is accomplished by constructing a recursive topology for transferring orders in a squad. This paper presents a concept of Sensing Troops, its architecture and a developed prototype for studying feasibility.

Key words Sensing Troops, IoT, environmental sensing, army hierarchy, scalability, configurability.

1. はじめに

身の回りのあらゆるデバイスをインターネットに接続する、IoT (Internet of Things, モノのインターネット) [1] [2] の時代が到来しつつある。従来のインターネットは、その末端は人であり、あくまで人と人を繋ぐための手段であった。一方IoTの世界では、人の手を介することなく、モノが主体となってインターネットに接続する。これにより、例えば家のドアそのものが施錠状況を監視し、鍵の閉め忘れを通知するといった実世界のモニタリングシステムや、窓そのものが外気温の低下を検出し、エアコンの運転モードを弱めるといった自律的な家電同士の省エネ連係システムなどが可能となる。さらにIoTの適用範囲はこのような宅内に限ったものではない。オフィスや都市全体、交通、農業、マーケティングなど、あらゆる分野に変革をもたらすコンピューティングパラダイムとして期待されている。

IoTにおける最も原始的かつ基本的な操作は、インターネッ

トを経由した「モノの状態取得」、及び「モノの操作」の2つである。これらは、インターネットを主体として見たときの、物理世界からの入力と物理世界への出力に該当する。先ほどのドア施錠監視システムの例の場合、「モノの状態取得」は、ドアによる鍵の施錠状態のセンシングと家による住人不在イベントの検出が該当する。さらに「モノの操作」は、インターネットを経由したドアの施錠操作、及びスマホやスピーカーによる閉め忘れの通知が該当する。家電に限らず、ペンや本、書類の一枚一枚といった、身の回りのあらゆるモノに対してこの2つの機能が満たされれば、インターネット（あるいはソフトウェア）と物理世界との障壁はなくなり、これまでない価値の創造が実現される。IoTの実現と普及のためには、この2つの機能をいかに達成するかが重要な鍵となる。

このうち「モノの状態取得」（以降、IoTセンシングと呼ぶ）を実現するためには、いくつかの課題を解決する必要がある。

第一にスケーラビリティの問題が挙げられる。あらゆるモノが自身の状態や周辺環境の状況を定期的にセンシングしてインターネットに送信するため、センシングデータを収集する基盤側へのアクセスの負荷集中が発生する。現在では、電源の供給やセンサデバイスの小型化の限界といった技術的な制約により、IoTにおけるモノとして接続可能なデバイスには限界がある。しかしながら、将来的には様々なモノがセンシング機能を持つことは想像に難くない。よってセンシング基盤には、モノの増加に対してうまく負荷を分散させ、スケールできる仕組みが必要である。

また、設定・管理の容易性という課題も考えられる。IoTセンシングにおけるモノの設定・管理には、新規デバイスを追加する際の初期設定や、デバイス故障・通信切断などの異常検出、上位アプリの要求に応じたセンシング頻度の変更など様々なものが考えられる。これらの設定・管理を全てのモノ一つ一つに行うことは非現実的であり、ある程度の自動化や、一括での設定変更といった工夫が必要である。

本研究ではIoTセンシングを実現する方法として、軍隊のアナログを取り入れたセンシング基盤、センシング部隊を提案する。センシング部隊では、まずIoTにおける個々の「モノ」を「兵（二等兵）」と見立てる。この二等兵は、自信の状態に関するデータ（起動状態や利用状況、本体温度など）や、周辺環境に関するデータ（温度や湿度、照度など）といった様々なセンシング結果を収集し送信する機能を持つ。この二等兵の一定のまとまりを分隊と見立て、分隊ごとにセンシング結果を集約するような階層構造（ヒエラルキー）を構築することで、負荷分散と高スケーラビリティを実現する。さらに上官から部下への命令の伝達というコンセプトを基盤に取り入れ、分隊ごとに自律的に二等兵を監視させることで、設定の容易化を達成する。

本稿では、センシング部隊のコンセプトと具体的なアーキテクチャ、及びその中で動作する各デバイス・アクタの関係や役割について検討する。また、ソフトウェアとRaspberryPiによるセンシング部隊のプロトタイプを作成し、システムの実現性の確認を行う。

2. IoTセンシングに求められる要件

本章では、IoTセンシングを実現するための3つの要件について述べる^(注1)。

2.1 高スケーラビリティ

どのような種類のデバイス（あるいはセンサ）をどのような範囲に、どのような粒度で配置し、どのような頻度でセンシングするかは、IoTの適用分野によって大きく異なる。例えば、農業における作物のモニタリング[3]や交通網の監視[4]のような分野では、均一の粒度で特定のセンサを数多くの配置することが多い。また、スマホ等を用いて不特定多数の利用者がセンシング結果を共有する参加型センシング[5]では、様々なセンサの情報が不特定多数のデバイスから送信される。ほかにも、適

用分野を特定せず、あらゆるセンシング結果をあらゆるアプリが横断的に共有するような利用形態を考えた場合、極めて多くのデバイスがセンシング結果を送信することになる。よってこれらを支えるIoTセンシング基盤には、数十～数百万オーダーでの接続が可能で、スケーラブルな接続トポロジを構成できる必要がある。

また、IoTデバイスとして接続できるデバイスは年々広がりつつある。宅内環境を考えた場合、IoTデバイスとして利用可能なものは、テレビや冷蔵庫、電子レンジなどの高機能かつそれ自体電源供給が必須なものに限られていた。近年では、ネットワーク制御可能な照明[6]や、カフェイン量やカロリーを計測できるコップ[7]なども製品化されつつある。このようにIoTデバイスの普及という観点からも、スケーラビリティの確保が求められる。

2.2 設定・管理の容易性

数多くのデバイスがIoTセンシングに参加する場合、それら個々のデバイスを全て手作業で設定・管理することは非現実的である。ここでの設定・管理には様々なものが考えられる。まず、新規IoTデバイスを追加する場合には、そのデバイスがどのような役割で、どのようなセンサを持つかといったメタ情報を収集し、管理する必要がある。センシングの頻度も常に一定とは限らない。スマートフォン等のバッテリー式のIoTデバイスでは、省電力化のためにバッテリーの残状況に応じてセンシング頻度を抑えるという、動的な設定変更も考えられる。さらに、数多くのIoTデバイスを用いる場合、その故障率も同時に増加するため、個々のデバイスの故障や通信切断などの異常検出といった管理も必要となる。これらのことから、IoTセンシング基盤には柔軟かつ低コストでの設定・管理が求められる。

2.3 異種混合環境への適応

個々のIoTデバイスがどのようなセンサを持ち、どのようなデータを送信するかは様々である。冷蔵庫であれば現在の消費電力や、庫内温度、総起動時間、総閉時間などがセンシングの対象であり、また窓であれば外気温と内気温、開閉状況などがセンシングの対象となる。IoTセンシング基盤には、これら異種混合デバイスの様々なセンシング結果を一元的に管理できることが求められる。

加え、IoTデバイスの性能にもばらつきがある。テレビ等の高機能なデバイス以外にも、先述のコップのような無線LAN環境に接続できる最低限のスペックしか持たないデバイスも存在する。よって、IoTセンシング基盤に参加するためのIoTデバイスの必須要件は、可能な限り軽量であることが望ましい。

3. センシング部隊のコンセプト

センシング部隊では、そのアーキテクチャや役割の名前に軍隊のアナログを導入することでIoTセンシングにおける諸問題の解決を図る。まず、センシング部隊のアーキテクチャを図1に示す。センシング部隊では、センシングデバイスである二等兵は軍曹をトップとする分隊としてまとめられ、その分隊を大尉をトップとする部隊としてまとめるヒエラルキー構造を成

(注1)：なお前提として、本稿では個々のIoTデバイスの電源供給、及びIPネットワークへの接続性は確保されているものとし、議論の対象外とする。

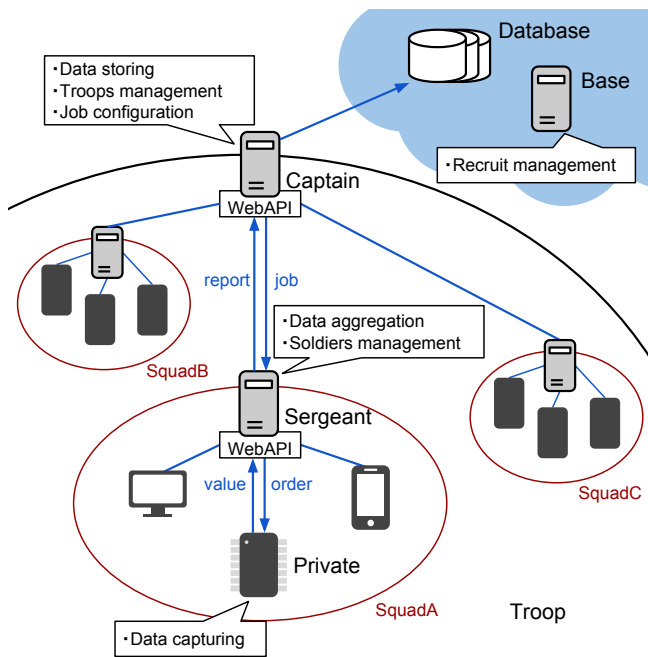


図1 センシング部隊のアーキテクチャ

している。またこの構造の外側には、新規デバイスの追加の仲介を行う募集所と部隊全体で収集したデータを格納するためのデータベースが存在する。以降では、このセンシング部隊を構成するアクタである二等兵・軍曹・大尉およびそのグループである分隊・部隊についての概要を示す。

二等兵 (Private) とは、実際のデータを取得するためのセンサを持ち、その値を取得するアクタである。二等兵はいずれかの分隊に所属し、上官である軍曹からの命令 (**order**) に応じた時間間隔でセンシングデータ (**value**) を軍曹に送信する。

軍曹 (Sergeant) とは、複数の二等兵をなんらかの意味的ラベルでまとめた分隊 (**Squad**) を管理するアクタである。軍曹は、分隊として受けたセンシングの指令 (**job**) を配下の二等兵に振り分け、それらの兵から送信されたセンシングデータを集約したレポート (**report**) を大尉に報告する。

大尉 (Captain) とは、複数の分隊をまとめた部隊 (**Troop**) を統括するアクタである。大尉の部隊長としての役割は、配下の分隊で集められたレポートをまとめてデータベースに送信することや、部隊全体のセンシング設定を配下の分隊に振り分けることである。また大尉は、これらの各種設定の手段や管理情報をユーザに提供する、インタフェースとしての役割を持つ。

募集所 (Base) とは、新規二等兵がセンシングに参加する際に大尉・軍曹と新兵との仲介を行うアクタである。入隊先の分隊がわからない段階の新兵に対し、大尉に問い合わせることで参加先の分隊を確認して伝える役割を持つ。

4. アクタとグループ

センシング部隊は、3. で述べたように二等兵・軍曹・大尉・募集所のアクタと分隊・部隊のグループで構成される。ここでは、この構成要素それぞれの機能や必要要件について述べる。

4.1 全アクタ共通

すべてのアクタはネットワーク的に接続され、それぞれのデータや命令などのやり取りは HTTP 通信で行われる。そのため、すべてのアクタはネットワークに接続することができ、かつ必要に応じた HTTP 通信を実行できる必要がある。

4.2 二等兵

二等兵は実際にセンサ値を取得するデバイスであり、その機能には次のようなものがある。

- F1:** 分隊 (軍曹) への入隊処理
- F2:** センシングデータの取得・送信
- F3:** heartbeat の送信/センシング命令の受信

具体的には、まず **F1** では、入隊する分隊 (軍曹) に自身の情報を送信し、軍曹から返された命令を読み込む。**F2** は二等兵の主要機能であるセンシングそのものであり、**F1** で取得した命令に応じて取得するセンサ値の種類やそのインターバルを設定し、取得したセンシングデータを軍曹に送信する。**F3** の heartbeat は、自身が正常に動作していることを示すために一定時間ごとに送信するもので、生存確認に加えて分隊の指令が変更された場合に命令を取得する役割もある。

これらの機能を実装するために、二等兵は次のような要件を満たす必要がある。

- R1:** センシング対象の値が取得できる
- R2:** HTTP の GET/POST が発行できる
- R3:** 軍曹へのネットワーク的な到達性がある

R1 はセンシングを行うために必須であり、これは環境センサの他にも IoT デバイスのバッテリー残量などのステータスも含まれる。**R2** は **F1-F3** の各機能を実行するために必要であり、**R3** はその通信の送信先に到達できなければならない、ということである。二等兵の通信はすべてリクエストでデータを送信して何らかのレスポンスを受け取る、というものであるため、HTTP のサーバ機能は必要とされず **R3** の逆向きの到達性は不要となる。また、二等兵で動作させるプログラムで行う処理は設定の読み込み・センサ値の取得・取得データの送信のみであるため、二等兵を実装するデバイスには CPU の処理能力はあまり要求されない。

これらの条件を満たすデバイスとしては、小規模なものでは PIC や AVR といったマイコンにネットワークモジュールとセンサを取り付けたものがある。また、高機能なものでは Raspberry Pi やスマートフォンといったものも含まれる。取得するデータによっては、通常のパソコンでも実装することができる。このように二等兵は様々なデバイスで実装することができるが、各種通信はすべて HTTP という共通のプロトコルにて行われるため、他のアクタから見たときの実装デバイスの差は持っているセンサのような属性の差のみである。そのため、分隊の中に様々な種類の二等兵が存在する異種混合の部隊を特別な対応や機能を持たずして実現することが可能である。

4.3 軍曹

軍曹は大きく分けて、分隊長としての機能と部隊の構成員と

しての機能が必要となり、実際の機能としては次のようなものがある。

- F1:** 新規二等兵の入隊受付
- F2:** 二等兵からのセンシングデータの受信・集計および大尉へのレポート送信
- F3:** 二等兵への命令の生成および通達
- F4:** 部隊（大尉）への入隊処理
- F5:** 大尉への heartbeat の送信/指令の受信

F1-F3 は主に軍曹の分隊長としての役割であり、**F4, F5** は部隊の構成員としての役割である。具体的には、**F1** は新兵の入隊志願メッセージに対して命令の通達および分隊長としての ID の割り当てを行う。**F2** では、配下のすべての二等兵からのセンシングデータを受信・集約したものを大尉へレポートとして送信する。**F3** では、分隊に対して与えられている指令から各二等兵に割り当てる命令を生成し、それを二等兵の入隊時および heartbeat へのレスポンスとして通達する。**F4, F5** は二等兵のものと同様で、送信先が大尉であることと受け取る指令が分隊に対するものであるという違いがあるのみである。

これらの機能を実装するために、軍曹は次のような要件を満たす必要がある。

- R1:** HTTP の GET/POST が発行できる
- R2:** 大尉へのネットワーク的な到達性がある
- R3:** HTTP のサーバ機能を持つ

R1 は二等兵のものと同様で、大尉との各種通信に利用するためのものであり、またこの通信が行えるためには **R2** の要件を満たす必要がある。**R3** は分隊長として二等兵からの通信を受けるために必要となる。

4.4 大尉

大尉は部隊長として分隊を管理する機能に加えて各種センシング設定やデバイス管理といった部隊全体の管理者としての以下のような機能が必要である。

- F1:** 新規分隊の加入受付
- F2:** 各分隊のレポートの受信・集計およびデータベースへの送信
- F3:** 分隊への指令の生成および伝達
- F4:** 部隊全体のセンシング設定の変更
- F5:** 各種設定・情報取得のための API の提供

F1-F3 は軍曹の二等兵に対してのものと同様である。**F4** のセンシング設定は、部隊の場合は上位アクタではなくユーザによって行われる。このセンシング設定のほかにも、大尉には配下にどんな軍曹・二等兵がいるのかといった情報や故障した二等兵の把握、また新規二等兵を追加する分隊の選択といった機能をユーザに提供する必要がある。**F5** はそのための API を提供するものであり、一般的な利用ではその API を扱うような GUI を実装する必要がある。

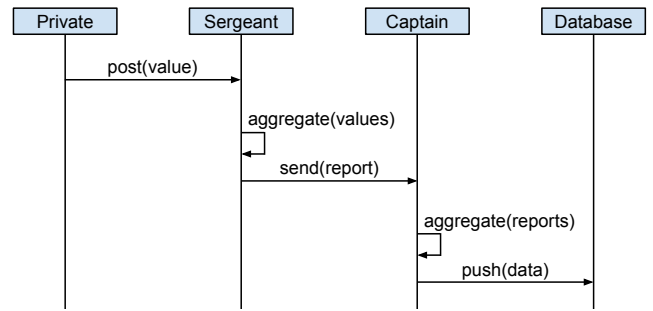


図 2 データ収集の流れ

これらの機能を実装するために、大尉は次のような要件を満たす必要がある。

- R1:** HTTP のサーバ機能を持つ
- R2:** 格納先のデータベースへの書き込みができる

軍曹と同様に配下からの通信を受け付ける必要がある、またユーザとの情報のやりとりのために **R1** が必要である。**R2** は最終的なデータの格納先が存在して書き込みを行う必要があるためである。データベースの所在は大尉自身でも外部でもよく、また fluentd などを経由してデータ書き込みを行うのであれば、大尉からデータベースそのものに直接アクセスできる必要はなくその中継サーバにアクセスできれば良い。

4.5 分隊・部隊

分隊は複数の二等兵をなんらかの意味的ラベルでまとめるものであり、たとえば環境センシングの場合では一つの部屋に 1 分隊を配置し、その分隊には「その部屋」というラベルを持たせるといったかたちである。分隊が受け取る指令はそのラベルに対応したものになり、この指令に応じて配下の二等兵にそれぞれ命令を振り分ける。

部隊はその複数の分隊を更にまとめるものであり、また、一人のユーザが管理するセンシングの最大単位としての意味合いも持つ。たとえば上記のように部屋ごとに分隊が存在する場合に建物一つに対して一つの部隊を配置するといったかたちになる。

5. 処理の流れ

今までに述べたアーキテクチャおよびアクタ・グループを踏まえて、センシング部隊による IoT センシングをセンシング結果の集約・命令通知・新規加入の三つのシーンに分け、それぞれシーケンス図を用いて解説する。

5.1 センシング結果の集約

実際のセンシングデータを取得する二等兵からデータベースまでデータが到達するまでの流れを図 2 に示す。データ収集においては、まず二等兵がセンシングデータを送信し (post)、それを軍曹が集約し大尉に送信する (aggregate, send)。そして軍曹からのレポートを大尉が更に集約しデータベースに送信する (aggregate, push) という流れにより、二等兵からデータベースまでセンシングデータが到達する。

図中ではそれぞれの動作が順に行われるように表記されてい

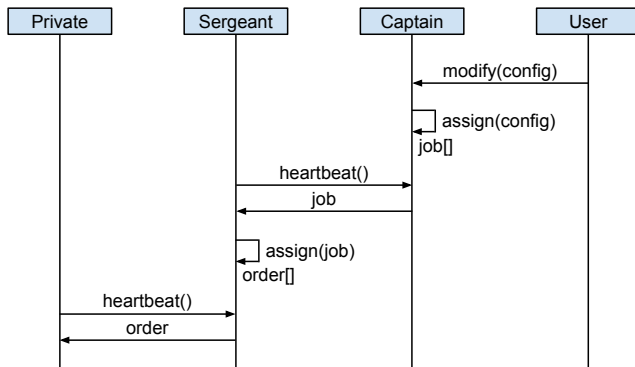


図 3 設定伝達の流れ

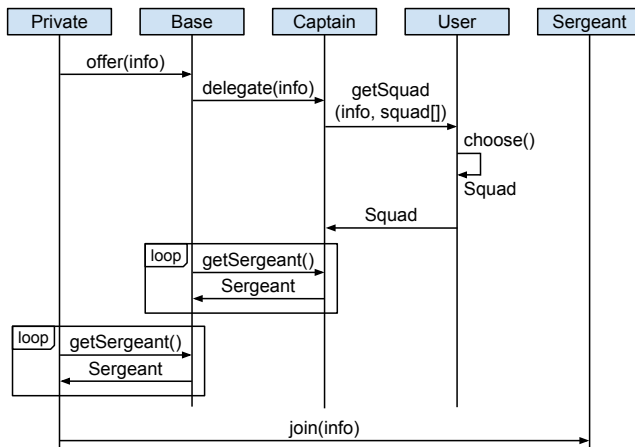


図 4 新規二等兵追加の流れ

るが、実際にはすべて独立したタイミングで動作する。軍曹は配下の二等兵からのセンシングデータがある程度キャッシュしておいてから集約・送信を行い、大尉も同様に配下の分隊からのレポートをまとめてデータベースに送信する。

5.2 命令通知

ユーザが部隊全体のセンシング設定を変更した際の各分隊、二等兵への指令・命令の伝達の流れを図 3 に示す。

図に示すように、部隊全体のセンシング設定を変更する場合、ユーザが大尉に対して設定変更処理を行う。設定の変更を受けた大尉は、その部隊全体の設定から分隊ごとの指令を生成し、分隊（軍曹）の次の heartbeat へのレスポンスに含めて通達する。分隊への指令に対しても同様に、新しい指令を受け取った軍曹は二等兵ごとの命令を生成し、二等兵の次の heartbeat へのレスポンスにて通達する。以上の処理を経て、ユーザによるセンシング設定の変更から大尉・軍曹を通して二等兵までの命令通知が行われる。なお、4. で触れているように二等兵および軍曹の heartbeat は設定更新に関わらず一定時間ごとに送信される。そのため、実際にユーザによる変更から二等兵へ命令が届くまでにはある程度のタイムラグが発生する。

5.3 新規加入

新しいセンシングデバイスを二等兵として部隊に参加させる際の処理の流れを図 4 に示す。

新兵追加の流れは、まず新兵が募集所に対して自身の情報を入隊申請メッセージとして送信する。次に、募集所が大尉対

して新兵の情報を大尉に送信し、どの分隊に所属させるかを問い合わせる。すると大尉がユーザに対し、新兵の情報と分隊のリストを提示して所属分隊の確認を行う。ユーザが新兵の所属分隊を決定すると、その情報が大尉から募集所を通して新兵に通知される。最後に、新兵が通知された分隊（軍曹）に対して入隊処理を行うことで新兵追加が行われる。

所属分隊の決定はユーザとの対話を含みある程度の時間がかかるため、一回の HTTP リクエストとそのレスポンスで手続き全体を完了することはできない。そのため、新兵は最初の入隊申請メッセージを送信した後、募集所に所属分隊の情報が返ってくるまで一定時間ごとに問い合わせを続け、募集所が所属分隊の情報を得られ次第問い合わせへのレスポンスにて分隊を通知する。

6. プロトタイプ

実際に動作するセンシング部隊を実装するには、4. で述べた各アクタの機能を実装する必要がある。本章では、5. で述べた利用シーンのうち、センシング結果の集約に注目したプロトタイプの実装を行う。なお軍曹・大尉はソフトウェアとして実装し、二等兵はソフトウェアとハードウェアのものを作成した。

まずソフトウェアの二等兵のプロトタイプでは、二等兵・軍曹・大尉が一つずつ存在しており、二等兵からのセンシングデータを軍曹がキャッシュし、それを大尉が更にキャッシュしたうえでデータベースに送信する、という流れになる。データの送信インターバルは二等兵<軍曹<大尉と上にいくほど大きな値に設定し、データは確実にいくつかキャッシュされるようにした。なお、二等兵もソフトウェアであるためセンシングする値はプログラム中で生成した乱数とタイムスタンプのペアとした。

ここで、作成したプログラムでの軍曹・大尉の WebAPI を以下に示す。

新規二等兵の加入

`http://sensingtroops/pvt/join`

センシングデータの送信

`http://sensingtroops/pvt/{id}/post`

命令の取得

`http://sensingtroops/pvt/{id}/order`

新規分隊の加入

`http://sensingtroops/sgt/join`

レポートの送信

`http://sensingtroops/sgt/{id}/send`

命令の取得

`http://sensingtroops/sgt/{id}/job`

なお、二等兵・軍曹・大尉それぞれのプログラムは python を用いて作成し、軍曹・大尉に実装したサーバ機能には python の簡易サーバ用モジュールを利用した。作成したプログラムは、WebAPI の呼び出しと HTTP サーバ機能をパッケージした自作モジュールが 87 行、それを利用した上で二等兵のプログラ

ムが95行, 軍曹が139行, 大尉が84行となった。これらのプログラムを利用して動作テストを実行したところ, データベースにセンシングデータが格納されていることが確認できた。

次に, 二等兵を実際のデバイスである Raspberry Pi で実装した。Raspberry Pi では LinuxOS が動作するため, python で作ったプログラムをそのまま動作させることができる。そこで, python で作成したプログラムのうちセンサ値の取得部分を CPU 温度^(注2) に差し替えたものを作成し, 分隊に参加させることでセンシングを行った。このプロトタイプでも問題なくデータを収集することができたため, ごく簡単な IoT センシングを実現できたと言える。

なお今回のプロトタイプではセンシング設定の割り振りと募集所は実装されておらず, 二等兵への命令や軍曹への指令は上官側で固定したものを利用し, 接続先の上官はプログラム中に書き込んだものを使用している。

7. 考 察

センシング部隊では, 2. で挙げた高スケーラビリティ, 設定・管理の容易性, 異種混合環境への適応という3つの要件に対して, それぞれ次のように解決している。

まずスケーラビリティに関しては, 軍隊のヒエラルキを基にしたデバイスの管理構造を採用することにより解決している。センシングデバイスを増やす場合にはそれに依って分隊(軍曹)を増やせばよく, 分隊の数も増えすぎの場合には別部隊として分離すれば良い。これらは階層的に管理されているため, たとえば二等兵が10,000台増えた場合でも, 1分隊に100台ずつ割り当てることで軍曹・大尉が直接管理するアクタの増加量は100に抑えられる。またデータの通信や処理の負荷に関しても, 大尉がすべてのセンシングデータを直接受信・処理するのではなく, 各分隊にそれらの負荷を分散させている。

次に設定・管理の容易性についても前述のデバイスの管理構造に基づいた管理を行うことで解決している。センシング設定を大尉から順に分隊・二等兵と自動的に割り当てていくことで, ユーザは部隊に対する設定を変更するだけで部隊全体の二等兵に命令を与えることができる。また新規センシングデバイスの追加に関しては, 末端のデバイス自体から実行する必要のある通信や設定を自動化し, ユーザはデバイスのネットワークへの接続と高級なインタフェースから分隊を選択するのみとすることで容易化している。

更に異種混合環境への適応に関しては, アクタ同士の通信を HTTP 通信で行うことで達成している。IoT センシングでは様々な種類の二等兵が想定されるが, センシング部隊ではデバイスの種類に依らない統一された通信手段を用いているため, 軍曹や大尉, ユーザから見た場合のデバイスの違いはそのメタ情報の違いのみとなる。また二等兵のプログラムでは, HTTP リクエストの発行を一定時間ごとに繰り返し, かつごく簡単な設定の解釈ができればよい。そのため, ごく最低限のスペック

しか持たないデバイスであってもセンシングに参加することができる。

このようにセンシング部隊では各要件に対して解決を図っているが, その設計によっていくつかの短所を持つ。まず, HTTP 通信そのもののオーバーヘッドがある。環境センシングなどで専用のデバイスを作成して通信する場合, ZigBee などの無線通信手段を用いて最小限のデータだけをやり取りすることが多く, またイーサネットを用いる場合でも簡単なセンシングデータの送信であれば UDP 通信でも可能である。これらの通信手段に対して HTTP の場合は TCP 通信のヘッダに加えて HTTP のヘッダも必要なため, 送信するデータ量に対してオーバーヘッドが非常に大きくなってしまふ。そのため, 専用のセンシングデバイスを開発するシステムに対しては通信量の面で不利である。また, センシング設定の更新のリアルタイム性が低いことが挙げられる。センシング設定の更新には問い合わせによる遅延が発生するため, 状況に応じた連続的な設定変更には対応は難しい。設定の問い合わせを行う heartbeat の間隔を短くすることで遅延を少なくすることはできるが, その分通信量や回数は増加してしまうため, 設定の応答速度は通信負荷とのトレードオフとなる。

8. おわりに

本稿では, 軍隊のアナログを取り入れた IoT センシング基盤であるセンシング部隊を提案した。ヒエラルキー構造のアーキテクチャの採用や HTTP という統一された通信方法の利用により IoT センシングにおける3つの要件を解決している。また, センシング部隊のアーキテクチャやそれを構成するアクタの仕様・動作を検討し, システムの特徴について考察を行った。

今後の課題として, 実際に部隊に割り当てられるセンシング設定やアクタ間でやり取りされるデータの形式の詳細な検討がある。

謝辞 この研究の一部は, 科学技術研究費(基盤研究 B 26280115, 15H02701, 若手研究 B 26730155, 萌芽研究 15K12020)の研究助成を受けて行われている。

文 献

- [1] D. Miorandi, S. Sicari, F.D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol.10, no.7, pp.1497–1516, 2012.
- [2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol.54, no.15, pp.2787–2805, 2010.
- [3] M.T. Lazarescu, "Design of a wsn platform for long-term environmental monitoring for iot applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol.3, no.1, pp.45–54, March 2013.
- [4] X. Yu, F. Sun, and X. Cheng, "Intelligent urban traffic management system based on cloud computing and internet of things," *International Conference on Computer Science Service System (CSSS)*, pp.2169–2172, Aug. 2012.
- [5] R.K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol.49, no.11, pp.32–39, Nov. 2011.
- [6] "Philips Hue," <http://www2.meethue.com/>.
- [7] "vessyl," <https://www.myvessyl.com/>.

(注2) : Raspberry Pi では `/sys/class/thermal/thermal_zone0/temp` を読み込むことで CPU 温度を取得することができる