Vol.2018-CDS-22 No.6
2018/5/31

スマートコントラクトを使った IoT システムの提案

古都哲生[†1]　峰野博史[†2]

概要：IoT システムにおいて，ブロックチェーン上で稼動するスマートコントラクトを利用するにあたり，効率的な実装方法について提案する．具体的には，IoT システムとブロックチェーンシステムの融合の際の必要な基本機能を明確化し，スマートコントラクトの特性，性能を評価した．また，その評価結果を元に，現状のブロックチェーンでの効果的な利用方法を明確化し，ブロックチェーンを使った IoT システムに必要とされる要件を明らかにした．

キーワード：IoT，ブロックチェーン，スマートコントラクト

Proposal of IoT System with Smart Contract

TETSUO FURUICHI[†1]　HIROSHI MINENO[†2]

Abstract: In the IoT system, we propose an efficient implementation method when using smart contract running on the BlockChain. Specifically, we clarified the basic functions necessary for fusion of the IoT system and BlockChain system, and evaluated the characteristics, performance and reliability in BlockChain. Based on the evaluation results, we clarified the effective use method in the current BlockChain environment and clarified the requirement for the IoT system for BlockChain.

Keywords: IoT, BlockChain Smart Contract

1. はじめに

IoT が工場，産業用途分野で実用化され，さらに家庭や健康を対象にした民生品に向けた発展がめまぐるしい．これまでの IoT デバイスは，低消費電力，小型化，軽量化，低コスト，等に重点を置いて開発されてきた．また，実用化と共に多様化が進むことで，システム構築の難易度が上がっている．近年国内外で IoT フレームワークが多く発表されているが，現状はゼロからの開発を行う場合も多く，開発コストが増大している．さらに，IoT 機器における情報セキュリティの脆弱性による DDoS 攻撃の被害が顕著になってきており，その対応も必要となってきた．実験システムから実用システムに移行していることで，長期間運用を目的としたシステムも重視されているが，半導体デバイスやモジュール等ハードウェアの供給期間の制限やソフトウェア脆弱性対応などで，組込み機器の売切り体制では要望を満たせない状況も発生してきており，これまで以上の可用性を配慮した IoT システム構築の難しさも明らかとなってきている．

1.1 IoT 構成の多様化

多くの目的に対応する IoT は，ハードウェア，ソフトウェア構成が多様化している．これまでセンサとコントローラで構成される軽量モジュールは，低価格，低消費電力に重きを置いていた．モバイルデバイスの発展で安価になってきた SoC(System on Chip)を利用することで現実的になってきた比較的潤沢なハードウェア・ソフトウェア構成を採用したモジュールは，エンドデバイス側でデータ処理を行うことができ，転送データ量を減らし，通信コスト削減することが可能となる．

また最近は，情報セキュリティ的な信頼性や可用性への要求も高まってきており，シンプルな機器においても PC やサーバシステムと同様の対策を求められるようになってきている．

1.2 ブロックチェーン技術の注目

暗号化技術が発展し，さらに暗号アルゴリズムを実用的な速度で実行できる潤沢なハードウェアが PC では一般的になってきたため，分散処理技術の一つである分散台帳システムのブロックチェーンが注目されている．すでにブロックチェーンを使った多くの仮想貨幣が発表され，本来の流通貨幣の利便性だけではなく，投機手段としての仮想貨幣が話題となっている．今回我々は，組込み機器の情報セキュリティ対策の手段として，分散台帳システムやスマートコントラクトの機能，性能に着目した．

1.3 IoT 分野へのブロックチェーンの適用

IoT は多様な種類のセンサデータを取得し，通信コスト削減のために，できる限りデータ量を減らし，低消費電力化のために，構成をシンプル化にしている．現在のブロックチェーンは，プロセッサ能力や潤沢なメモリリソースを

†1 イークラウド・コンピューティング／静岡大学 大学院 自然科学系教育部 情報学専攻
e-Cloud Computing&Co. / Graduate School of Science and Technology, Shizuoka University

†2 静岡大学学術院情報学領域／情報学部／総合科学技術大学院 情報学専攻
Graduate School of Science and Technology, Shizuoka University

使って，ネットワーク全体へのデータ共有と信頼性確保の為の分散化を行っている．そのため，これら二つは，軽量化と重厚処理の相反する考え方を持っている．しかしながら，今後の IoT の需要が増えると共に，セキュリティの強化や，システムの信頼性向上，また構成ハードウェアの継続的維持の要求が高まり，新しい解決手段が必要となってきた．そこで我々は，中長期の運用を目指している小中規模用途や，継続的に IoT システムを構築し，随時適応していく用途，システム障害に対する堅牢性を追究する用途，継続的なセキュリティ担保が必要な用途を対象とした，IoT センサモジュール製品を想定・企画し，そのプロトタイプシステムを構築した．

2. 従来研究と課題

ここでは，今回の対象とした IoT 技術とブロックチェーン技術を紹介する．

2.1 IoT

スマートフォンの誕生でセンサが低価格化し，クラウドコンピューティングの発展でインフラが低コストになり，さらに SNS の進化でデータが増大する好条件のなか, 2012 年のドイツの Industrie 4.0 や，その他日本を含めた各国の名乗りで，IoT はすでに実用化，具体化段階に入ってきた[1]．2017 年の国内 IoT 市場におけるユーザ支出額の実績は 6 兆円越えとなり，2022 年には 12 兆円に達すると言われている[2]．一般的になってきた IoT は，人，モノを含めたあらゆるモノをネットワークにつなげ，新しい価値を生み出し始めている[3] ．

(1) IoT プラットフォーム

近年 IoT プラットフォームは，国内外で数多く提唱されている[1]. アプリケーション開発や，通信のセキュリティ化，データ連係に関わるプラットフォームが多く発表され，実用化されている．最近では，LTE, LPWA, Wi-Fi, BLE, ZigBee 等の無線通信手段別のアプローチも盛んである．

また，2016 年秋に被害を出し，DDoS 攻撃の元となった MIRAI Bot Net が，IoT デバイスを使った被害拡大の火だねとなり，最近では新たな亜種による被害も報告されており，世間での IoT プラットフォームでの情報セキュリティ対策も期待されている．

(2) IoT システムの構成概要

IoT システムは，①センサ/アクチュエータ，②前処理，③通信，④蓄積，⑤分析．⑥推論，等の機能部分に分けられる．当初の IoT システムは，センサ，通信に特化していた．これらは，低消費電力，小型化，低コスト化，高信頼性，のために，ワンチップ・マイクロコントローラを使って構成され，OS なしもしくは RTOS(Real Time OS)で実装される場合も多かった．通信手段の多様化・大容量化・低コスト化に加え，低価格・低消費電力・高機能の SoC の発展や情報セキュリティの切り口での高信頼性要求により，Linux を実装した高度な IoT システムも実用レベルになってきた．この Linux モジュールは，IoT システムに要求される機能部分のほとんどをエンドデバイス側で実行することができ，ユーザの要望に併せて，さまざまなバリエーションの構成を実現することが可能になってきた．また，今までクラウドサーバ側実行していたさらにプロセッサ負荷のかかる分析や認識，推論・予測，等のデータ分析や AI による高度な処理も，Linux モジュールで実行可能となってきており，エンドデバイス側への要求はさらに高まってきている．

(3) IoT システムの要求用件

これまで述べたように，現在実用化されている IoT システムは用途に合わせて設計，開発を行っている場合が多く，様々な種類がある．センサの種類や，その情報の用途により，通信レイテンシーや伝送帯域で総称される通信速度が IoT サービス満足度に対して大きな要素となることが多い. また，通信コストを下げるために，独自回線でなく，インターネットを使って情報を伝送することも多く，情報セキュリティに対する対策も必要となってきている．

情報セキュリティに関しては，一般的に言われている三大用件である，機密性，完全性，可用性が重要であり，IoT に向けても同様の要件が求められている．先に述べた IoT 向けのマルウェア対策のためにも，アクセス権の確保，改ざん防止，非常時の対応が重要になってきている．しかしながら，数が多くなるシステムにおいては，導入コストも高額になり，モジュールのコストダウンの要求も強く，通信速度や情報セキュリティ強度とのトレード・オフも随時行われているのが現状である．

2.2 ブロックチェーン技術の発展

ブロックチェーンは，ネットワークにおいて参加者による分散型合意形成を可能とし，すべての取引履歴を追跡可能にした分散台帳システムである．

これまで，ブロックチェーンを利用した多くの種類の仮想貨幣が流通しているが，ここでは最も有名なブロックチェーンである Bitcoin や，アプリケーション応用ができるブロックチェーンシステムである Ethereum を紹介する．

(1) Bitcoin

2008 年に Satoshi Nakamoto と名乗る人物により投稿されたブロックチェーン技術に基づいて, 2009 年に運用が開始された Bitcoin が仮想貨幣，ブロックチェーンとして有名である[4]．Bitcoin は，Bitcoin クライアントと呼ばれるブロックチェーンノードと Bitcoin ネットワークから構成されるシステムである. Bitcoin クライアントが発行する取引情報がトランザクションとして，Bitcoin ネットワークに送りこまれ，Bitcoin クライアントの一種であるマイナーがマイニング(採掘)することで，ブロックとして Bitcoin ネットワークから承認され，取引が成立することで成り立

っている[5].

(2) **Ethereum**

Ethereum は，2013 年 11 月に Viralik Buterin 氏のホワイトペーパーにより提案されたスマートコントラクトによるアプリケーション構築を可能としたブロックチェーンである．Bitcoin が，暗号通貨の所有権の移動に特化しているのに対して，Ethereum は暗号通貨の移動だけでなく，分散アプリケーションを自由に構築できることを特徴としている[6][7].

2.3 ブロックチェーンの技術

ブロックチェーンは，分散型ネットワーク上で，やりとり情報であるトランザクションを分散して管理する「分散台帳」を実現した，分散型データベースとも言える．それらのトランザクションをまとめたブロックと呼ばれるシーケンシャルなデータのリストを，複数のノードで管理・運営しする．また，分散合意形成アルゴリズムを用いたマイニング(採掘)処理により，ブロックの正当性を担保する．現在の分散合意形成アルゴリズムは，PoW(Proof-of-Work: 仕事量の証明)が主流であり，ネットワークの構成ノードの同意の下で，Difficulty 値(難易度)が設定され，マイニング時間を調整する仕組みを持っている．Bitcoin を例とすると，トランザクションのブロック化は約 10 分で完了する設定となっている．分散合意形成アルゴリズムは，PoW の他に，PoS(Proof-of-Stake: 保有による証明),や PoI(Proof-of-Importance: 重要性の証明) と呼ばれるプロセッサ資源や電力を費やさない方法も検討され，導入例も増えてきており，今後はさらに利用されると考えられている．

(1) **ブロックチェーンの動作**

ブロックチェーンは，ノード，P2P ネットワーク，トランザクション，ブロック，分散台帳，マイニング，と様々な要素で構成されている．

ブロックチェーンは物理的な通常のネットワーク上に構成された仮想的なネットワークである．その接続単位をノードと呼び，ノード同士は，一対一(P2P)で物理的に接続されている．それぞれのノードは，それぞれ固有の非対称鍵を持っている．ノードが発行する取引情報(トランザクション)は，ノード自身の秘密鍵で署名され，別ノードの承認後，P2P を介してブロックチェーンネットワークに拡散される．

拡散されたトランザクションは，あらかじめ設定された時間でまとめられて，マイニングとして PoW で代表される分散合意形成アルゴリズムを使ってブロック化される．そのブロック情報も複数承認過程を経て，分散台帳の情報として扱われる．

図 1 は，ブロックチェーンノード間での処理の流れの例を示したものである．まず Node #0 が取引のトランザクションを発行したとすると，その情報は Node #1 に渡り，承認された後，Node #2, #3, #5→#4,#6,#7 と拡散される．そこで，マイニング機能を持つ Node #6 が分散合意形成アルゴリズムを使ってマイニングを行うことで，ブロック化する．そのブロックの情報は，Node #5 経由でブロック情報を拡散され，ブロックチェーン全体で承認され，各ノードで有効な分散台帳として扱われる情報となる．

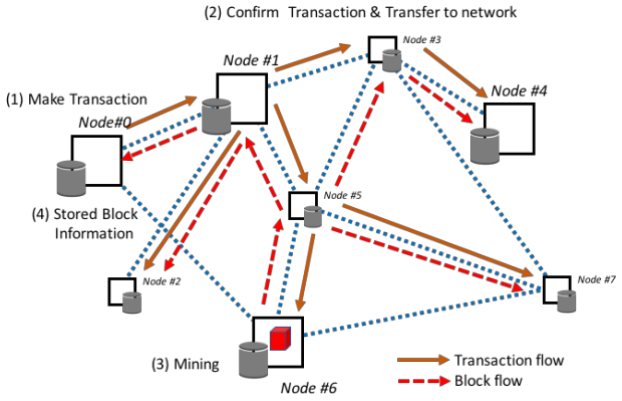

図 1 ブロックチェーンノード間での処理の流れ

(2) **スマートコントラクト**

Bitcoin が仮想貨幣の取引に特化した仕組みであるのに対し，仮想貨幣の取引だけでなく，ブロックチェーン上で共有化される一種のプログラムであるスマートコントラクトを扱えるブロックチェーンがある．Ethereum はその一つで，仮想貨幣の取引を主として，ノードがアクセスでき，仮想的なプログラムを実現することが可能である．

2.4 IoT 向けブロックチェーン

ブロックチェーンの分散管理の利便性と，その仮想貨幣の特徴に着目し，IoT への適応の期待度が上がってきている．電力系の分野において，IoT のアップデートを継続的におこなう事をブロックチェーンで効率的に行っている例もある[10]．また，IoT に対応するため，ブロックチェーンのクライアントプログラムにラッパーをかぶせ，ブロックチェーンとは別のネットワークも併用することで，ブロックチェーンの不得意なデータ転送をまかなう仕組みも開発され，実際に電力系のシステムで使われている[8]．さらに IoT をスマートホームで利用するために，ブロックチェーンを使った研究もされており，機密性，完全性，可用性を実現し，セキュリティとプライバシーのメリットが，ブロックチェーンのオーバーヘッドより大きいことを主張している[9].

3. ブロックチェーン IoT システムの実現

IoT システムにおいては，対象ネットワークに対して，IoT デバイスの登録やデータの受渡は重要な機能である．本研究では，ブロックチェーンのセキュリティ性に着目し，すでに使われている仮想貨幣 Ethereum のブロックチェー

ンクライアントプログラムである geth を利用し，IoT モジュールインターフェースの基本的な機能を持つ API を開発し，ブロックチェーン IoT モジュールに実装した．今回の研究では，既存のインターネットを利用した組込みシステム元に構築するため，小・中規模の IoT センサネットワークシステムを想定し，長期運用，拡張性や高信頼性を目指した．また，パブリックな仮想貨幣の価値の問題を回避するために，プライベートなブロックチェーン上に構築した．

3.1 基本要件

小・中規模で長期間稼動可能な実用的な IoT システムに必要な要件としては，複合的な連携や柔軟性，接続性，拡張性，セキュリティ性など項目が挙げられる．今回の基本要件として，①汎用ネットワークの利用，②汎用ハードウェア・OS の利用，③接続可能センサモジュールの事前登録，④故障，異常時の敏速対応，⑤ネットワークへの安全なアクセス，と定義した．特に③～⑤の要件を考慮し，ブロックチェーンの仮想貨幣を利用した管理用スマートコントラクトとしてプロトタイプシステムを構築した．

3.2 ハードウェア・ソフトウェア構成

Ethereum の geth プログラム は，Go 言語で実装されており，x86 または ARM プロセッサの Linux/Unix 環境で稼動する．システムは TCP/IP ベースのネットワークで接続され，その物理的な通信方法は，有線，無線を問わない．それぞれの機器(以下「Node」)は固有の IP アドレスを割り当てる必要がある． Node は geth プログラムとセンサとのインターフェースや外部のクラウドサーバとの通信機能も持たせたアプリケーションを稼動できるようにした．また，センサの扱いや外部ネットワークへの通信権限は，ブロックチェーンを利用した Node 間接続とは別ポリシーとし，IoT 機器としての接続性の柔軟性を確保した．

3.3 センサデータの流れ

固有の IP アドレスを持ったそれぞれの Node は，Node 上で実行される geth プログラムで相互に接続することでブロックチェーンを形成する．Node はそれぞれがブロックチェーンにアクセスするためのアカウントである address を持っており，Node 間の通信はトランザクションとして address 間で行われる．今回のシステムは，そのブロックチェーン内に存在するミドルウェア的位置付けのスマートコントラクトに IoT 管理 API 機能を持たせ，ブロックチェーンネットワーク全体で IoT システムとして稼動するようにした．

図 2 は，今回開発したブロックチェーン IoT モジュールを使ったシステムの一例の処理の流れを示したものである．センサを持つノード(Sensor Node)からブロックチェーンネットワーク経由で，ゲートウェイ機能を持つノード(GateWay Node)を介して，クラウドサーバにデータを送り出している．

具体的には，Sensor Node において，①IoT Application が Sensor Library 経由で，接続されているセンサから情報を読み出す．②IoT Application が，BlockChain Client (geth)とネットワークを経由してブロックチェーン上のスマートコントラクトにデータを渡す．③データを受け取ったスマートコントラクトは，受け取った旨のイベントを GateWay Node のネットワークと BlockChain Client(geth)を介して，GateWay Node 側の IoT Application に送り出す．④GateWay Node 側の IoT Application は，スマートコントラクトから格納されたセンサデータを読み取り，⑤ IoT Library を使ってクラウドサーバにデータを送り出すこととなる．

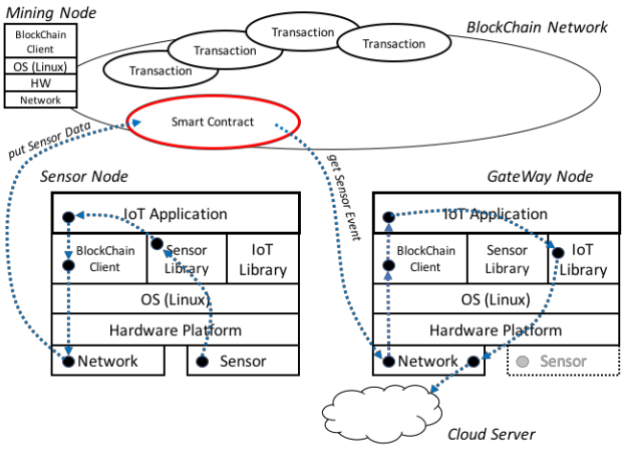

図 2 スマートコントラクトとノード間の処理の流れ

この接続で特徴的なことは，Sensor Node と GateWay Node は，接続するブロックチェーンの固有 ID と，接続するスマートコントラクトの address だけの情報で，情報のやり取りができることである．これは，双方の Node がお互いの IP アドレスがわかっていなくても通信できるということである．また，スマートコントラクトの実行は，マイニングされるハードウェアで仮想的に実行されることになり，この例では，Mining Node が実行マシンとなる．なおこの Node は複数個あっても構わない．

3.4 IoT スマートコントラクト

これまでの IoT デバイスの認証・データ受渡しは，IoT デバイス自身が認証機能を持つことで通信を行う方法や，特定された管理サーバと通信する方法などで実現されてきた．今回のシステムは，IoT のモジュール認証機能とデータ受渡機能を，ブロックチェーンのスマートコントラクトで実現しており，これまでの IoT デバイスのインターフェースとは異なった実装となっている．

一般的な IoT 機器は，センサ機器にユニーク ID や非対称鍵を持たせ，その ID や公開鍵を通信ノードやサーバ側で認証することで，機器接続の認証承認を行っていた．また，データ通信に関しては，VPN や SSL を使って通信線路を暗号化する方法や，送り側で暗号化＋署名化する方法

などが一般的である．

ブロックチェーンクライアントソフト経由でブロックチェーンに接続することで，アカウント address が相互承認されたユニーク ID となり，そのアカウントでやり取りされたデータはトランザクションとして扱われ，アカウントアドレスで署名されたことと同等の扱いとなるため，ブロックチェーンでのインターフェースは，データ暗号化以外の要素を含むこととなる．

また，今回の IoT スマートコントラクトは，IoT の為の仮想ネットワークと接続するための分散管理 API となっており，あらかじめトランザクションとしてブロックチェーンに登録，認証してある．さらに，今回開発したブロックチェーン IoT モジュールは，接続するブロックチェーン情報はあらかじめ読み込まれており，物理的に TCP/IP ネットワークに接続し，設定されたブロックチェーンに参加するだけで，稼動しているブロックチェーンと同期化し，その登録されたスマートコントラクトにアクセスすることができるようになる．

本 IoT スマートコントラクトは，大きく分けて 2 つの機能を持っている．一つ目はセンサ自体のマネージメント機能であり，もう一つはセンサデータの受渡機能である．IoT スマートコントラクトは，あらかじめ設定された Owner Node アカウント address でブロックチェーンにトランザクションとして登録することで，スマートコントラクトとして利用可能になる．その Owner Node のアカウント address は登録された IoT スマートコントラクトの特権ユーザとなり，操作権限を持つことになる．また，センサ登録は，その Owner Node アカウント address で行われる．その他，Owner 属性でしか制御できない機能がいくらかある．

本 IoT スマートコントラクトにおいて，既存の制御プログラムと比べて大きく違っている点としては，機能の呼出実行において，明示的な仮想貨幣によるコストを意識していることにある．各ノードからスマートコントラクトへの参照，設定，機能の実行は，本ブロックチェーンで規定している仮想貨幣が使われる．これは，仮想貨幣の残高が足らないと IoT スマートコントラクトへのアクセス自体ができないことであり，第三者が作ったマルウェアプログラムは，仮想貨幣を持っていないため，IoT スマートコントラクトを稼動させることができなく，ハッキングの大きな抑止力になる．

(1) センサデバイスの登録

IoT 管理機能の 1 つとして，接続可能なセンサデバイスの登録を行う．この設定は，スマートコントラクト作成者である Owner が行う事としてある．Owner は，接続可能な Sensor Node のアカウントアドレスを，登録関数の引数として設定し，トランザクションを発生させることで，スマートコントラクトに登録する．

(2) センサデバイスのアクティベーション

スマートコントラクトに登録している Sensor Node は，自身のアカウント address を使って，スマートコントラクトにアクセスし，自身のセンサデータチャンネルをアクティベートする．アクティベーションは，仮想貨幣の費用がかるようにプログラムされている．また，設定の際に，バッファリングできるデータ個数や，センサデータやり取りの仮想貨幣を使った費用を設定できるようにしてある．

(3) センサデータの受渡し

Sensor Node がセンサデバイスから受け取ったデータを IoT スマートコントラクトに登録する場合も仮想貨幣の費用がかかる様にしている．この仕様もまた，IoT システムへの不要データ登録を抑止させる効果が期待できる．今回は，小容量のデータ受渡を想定しているため，直接データをブロックチェーンにトランザクションとして扱うことにしてある．センサが大容量のデータを出力する場合は，ブロックチェーンでのトランザクションの負荷が大きいため，別途データ受渡の機構を持つべきである．

今回の実装においては，Sensor Node からのセンサデータが予め設定された個数スマートコントラクトに蓄えられると，スマートコントラクトはイベントと呼ばれる事象を発行する．予め GateWay Node は，そのイベントを geth プログラム経由で受信できるようにしておき，イベントを受け取った場合，スマートコントラクトに対して，データの引取を行う．このアクセスは，あらかじめ Sensor Node がアクティベーション時に設定した金額の仮想貨幣の支払いを要求しており，この設定も，権限，許可なしに無節操なデータアクセスを阻止する効果が期待できる．

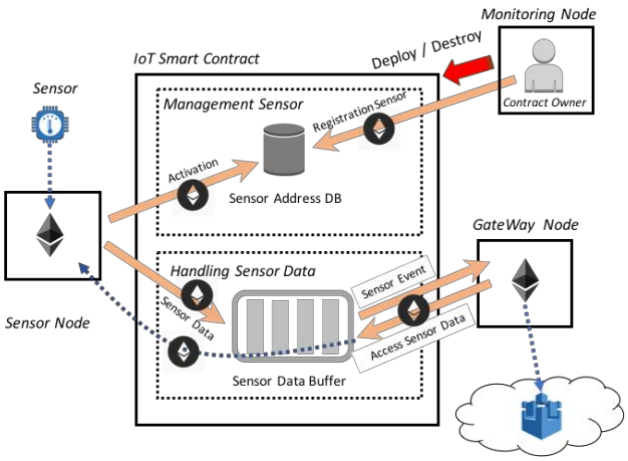

図 3　IoT スマートコントラクトの構成

4. プロトタイプ実装と評価

3 で検討した構成モデルの利用条件を明確にするために，実際に IoT システムと既存ブロックチェーンを組み合わせ，システムとしての傾向を評価した．特に IoT システムとしてみた場合の，センサデータ転送の際のレイテンシー時間

と，IoT モジュール側の主メモリの利用状況と，ディスクの利用状況は，システムの長期間の安定利用に係わるため，着目した．

4.1 評価に使った環境

図 4 は今回評価に利用した環境の構成図である．今回は Ethereum の Go 言語で実装された geth プログラムを，Desktop PC, Note PC，対象プロダクトに実装し，IoT システム用に Ethereum Private Net を構築した．さらにブロックチェーンネットに IoT API である IoT スマートコントラクトを実装し，予めトランザクションとしてブロックチェーンに生成しておいた．

評価に使う Node としては，コントラクト Owner である Monitoring Node，ブロックチェーンにセンサデータをアップロードする側の Sensor Node，センサデータをブロックチェーンから取り込む GateWay Node を用意し，それぞれブロックチェーンのアカウント address を割り当てた．また，それぞれの address には予め Private-Net の仮想貨幣を適当な金額を定め振り込んでおいた．

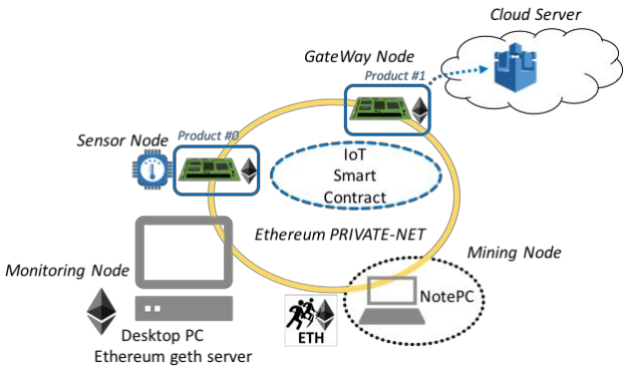

図 4　評価環境の機器構成

4.2 評価方法

評価は，各 Node で geth プログラムを稼動させブロックチェーンとリンクすることで行い，それぞれの Node での操作は評価用スクリプトを作成した．また，プロセッサ物理メモリ使用量，使用ディスク容量は，測定用のスクリプトを実行と同時に稼動させることで，リアルタイムのデータを取得するようにした．ブロックチェーン内トランザクションのブロック化のためのマイニング処理用の Mining Node は，ネットワークに接続された PC で実現した．

4.3 評価内容及び結果

(1) データ転送レイテンシー時間

本評価は，Sensor Node のセンサデータを IoT スマートコントラクトに送り出し，IoT スマートコントラクトが Event を生成し，GateWay Node がそれを受け取るまでの時間を測定した．評価は 100 回のアクセスを行った．図 5 にデータ転送のレイテンシー時間のヒストグラムと分布曲線を示す．レイテンシー値は 2.3 秒から 73.4 秒まで広範囲に分布しており，平均値は 27.0 秒であるが，その値より超える場合も多い．

(2) プロセッサ物理メモリ，ストレージ使用量の変化

本評価は，IoT デバイス上においてブロックチェーンクライアントプログラム geth がプロセッサ物理メモリとストレージをどれくらい使用するのかを測定するもので，特に IoT デバイスとしてデータを受渡している状況に着目した．

図 6 に IoT ブロックチェーンを使った IoT データ受け渡し時の，物理メモリ量とストレージの差分を示す．X 軸は処理の時間の流れで，左側の Y 軸はプロセッサ物理メモリ量の差分で，赤色の点線で示されるグラフとなる．右側 Y 軸はストレージ量の差分であり，濃紺の実践で示されるグラフとなる．

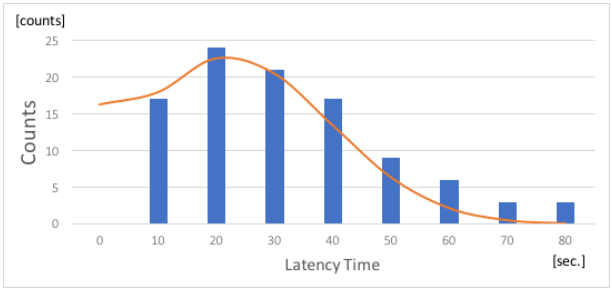

図 5　データ転送のレイテンシー時間の分布

評価のシナリオとしては，①geth の起動，②マイニング開始，③IoT スマートコントラクトとやり取りする Sensor Node のアカウント address のアンロック(パスフレーズでアカウントが取引可能状態となる)，④センサデータを IoT スマートコントラクトに送る Event を受け取る(複数回の繰り返し)，⑤マイニングの停止，となっている．

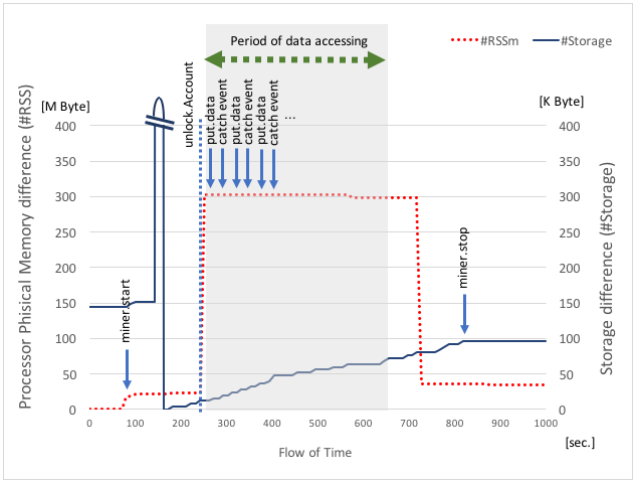

図 6　プロセッサ物理メモリ，ストレージ使用量の変化

①の geth プログラムの稼働後，②のマイニングが開始されると，約 20M Byte 程プロセッサ物理メモリの使用量が増加した．また，同時に，約 15 秒間隔でストレージが約

4K Byte 程度増加しはじめた．今回の測定では，geth プログラムによる何らかの最適化動作が実行された様で，ストレージの量がマイニング直後に一旦大幅に減少した．

③④の IoT スマートコントラクトとのやり取りが開始し始めると，プロセッサ物理メモリとして約 300M Byte 増加し，やり取りが終了した後，約 100 秒後に約 260M Byte の物理メモリの解放が行われた．その後⑤のマイニング停止で，ストレージ量の増加も無くなった．

4.4 結論

今回の評価において，開発した IoT スマートコントラクトは IoT デバイス API として正常に機能しており，その実行性能もリアルタイム性能は及ばないが，低レイテンシーの用途で，ブロックチェーン操作を工夫することで，IoT システムとして運用できることを確認できた．

評価の際，Sensor Node の address を変えてアクセスを試みたが，IoT スマートコントラクトは登録アドレスを正しく認識しており，該当する address でないとアクセスを許可しなかった．また，Sensor Node の address の仮想貨幣残高を少なくした場合も，スマートコントラクトに対するトランザクションが成立しなく，第三者の address からのアクセスがあっても，仮想貨幣が足らない場合は，アクセスできないことが確認され，情報セキュリティの観点でも有効に動作していることがわかった．

可用性に関しては，IoT デバイスをブロックチェーンのノードとして扱っているため，対象となる IoT デバイスが故障した場合も，新たなデバイスにブロックチェーンの設定と，アカウント address の生成さえ行えば，すぐにブロックチェーンへの接続が可能となり，代替えセンサとして利用できる．

(1) データ転送レイテンシー時間に関する考察

Ethereum のマイニング周期は 15 秒に設定されているため，データ転送のレイテンシー時間はその周期に同期すると予想していたが，実際にはさらに時間がかかってしまう場合が多かった．評価では 1 分以上時間がかかっている場合もあったため，秒単位のデータのやり取りには向かない．また平均レイテンシー時間は 30 秒弱であるが，長めの時間にばらつく場合もあり，ブロックチェーンを利用した IoT システムの場合は，数分のレイテンシーでも正常に稼動する設計を必要とする．

(2) ブロックチェーンクライアントの実行負荷の考察

Ethereum の geth プログラムは，その動作により動的なメモリ管理が行われており，約 300M Byte 単位でのメモリの確保，開放が行われている．今回の結果では現れていないが，評価の過程で最大 700M Byte のメモリ確保の時もあった．特に，アカウントの生成や，アカウントのアンロック，トランザクションの発行時等の，暗号アルゴリズムが利用されると思われる状況でのメモリ使用量増大は顕著である．従って，組込み機器において主記憶メモリ容量と，並列して実行されるプログラムとのメモリ容量共存には注意が必要である．

また，ブロックチェーン特有のマイニングは，トランザクションの実行に重要な役割があるが，その都度ストレージが約 4K Byte 増加している．このため，長期間の機器運用には，ストレージ容量の余裕度を持たせる必要がある．また，組込み機器で半導体メモリをストレージとして利用している場合は，書き込み回数制限の注意が必要である．対処方法の 1 つとして，マイニングが行われないとストレージ消費は行われないため，IoT 利用状況に応じて，マイニングを制御することや，一定周期で，ブロックチェーンの再構築をすることで，現実的なシステムとなると考えられる．

5. おわりに

今回 IoT スマートコントラクトを構築するために既存のブロックチェーンである Ethereum のクライアントプログラムである geth を利用した．本来 geth プログラムは組込み機器での実行は想定されていないと思われるが，その制約事項がわかった上で，適切な運用を行うことで，現行の組込み機器を使った IoT システムでも円滑に稼動し，特別なセキュリティ対策コードを書かなくても，情報セキュリティ的に安全なシステムを構築することができる．また，仮想貨幣のやり取りを伴うプログラムの実行権限は，これまでのプログラムと比べて非常にユニークで，IoT システムのセキュリティ対策の 1 つとしては有効である．

ブロックチェーンとのインターフェースは，予めブロックチェーン環境の構築や，それぞれの Node へのアカウント address を作っておかないと行けないこと，スマートコントラクトを稼動させるために，予めアカウントに仮想貨幣を渡しておく必要がある点，等，最初の負担は大きい．しかしながら，それらの対応は本質的な情報セキュリティ対策であり，ゼロから実装するには多くの作業工程が必要となってしまう．IoT における情報セキュリティ脆弱性対策の為にも，Security by Design の一手法として，ブロックチェーンを利用したシステムは有効であると考える．

参考文献

[1] 八子知礼他, "IoT の基本・仕組み・重要事項が全部わかる教科書", SB Creative, 2017.
[2] "国内 IoT 市場 産業分野別／ユースケース別予測、2018 年～2022 年", IDC Japan Press Release, https://www.idcjapan.co.jp/Press/Current/20180314Apr.html, 2018 年 3 月 14 日.
[3] Dave Evans, "The Internet of Everything How More Relevant and Valuable Connections Will Change the World", Cisco IBSG, 2012.
[4] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", https://bitcoin.org/bitcoin.pdf, 2008.
[5] アンドレアス・M・アントロノプス著，今井崇也・鳩貝淳

一郎訳, "ビットコインとブロックチェーン　暗号通貨を支える技術", NTT 出版株式会社, 2016.
[6] 田篭照博, "堅牢なスマートコントラクト開発のためのブロックチェーン［技術］入門", 株式会社技術評論社, 2017.
[7] コンセンサス・ベイス株式会社共著, "イーサリアムへの入り口", 日本電気株式会社, 2017.
[8] M. A. Walker, A. Dubey, A. Laszka, and D. C. Schmidt, “PlaTIBART: a platform for transactive IoT blockchain applications with repeatable testing,” in 4th Workshop on Middleware and Applications for the IoT (M4IoT), December 2017.
[9] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. "Blockchain for IoT security and privacy: The case study of a smart home", In Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. IEEE, 618‑623.
[10] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. IEEE Access, 4:2292--2303, 2016.