Vol.2020-GN-109 No.7
Vol.2020-CDS-27 No.7
Vol.2020-DCC-24 No.7
2020/1/23

高齢者支援サービスに向けたネットワークの QoS 制御

岡部寿男 [1,a)]　小谷大祐 [1]　上原亜矢 [1]　田中卓 [1]

概要：日欧共同研究プロジェクト ACCRA では，アジャイル型共創による高齢者補助ロボット用ネットワークプラットフォーム技術を研究開発している．ACCRA では，高齢化社会を見据えた補助サービスにおいて，家庭の様々な IoT デバイスによって多種多様のトラフィックフローが生成される．本研究では，デバイス，サービス，利用者の特性に応じてそのトラフィックを QoS(Quality of Service)制御するネットワーク制御システムと API を開発した．

キーワード：QoS　帯域保証　SDN　IoT

QoS Network control for elderly support services

YASUO OKABE[1,a)] DAISUKE KOTANI[1] AYA UEHARA[1] TAKU TANAKA[1]

Abstract: The European-Japanese joint research project ACCRA is researching and developing network platform technology for elderly assistive robots by agile co-creation. In ACCRA, a variety of traffic flows are generated by various IoT devices at home in an auxiliary service with an aging society in mind. In this study, we have developed a network control system and API that control QoS (Quality of Service) according to the characteristics of devices, services, and users.

Keywords: Qos bandwidth control method SDN IoT

1. はじめに

IoT（Internet of Things）対応のサービスには，スマートシティ，製造，輸送，その他の多くの分野など，さまざまな分野で人々の生活を向上させる大きな期待が寄せられている[1]．スマートホーム[2]は，IoT のアプリケーションとして注目を集めている分野の 1 つである．スマートホームの用途は，エネルギー管理[3]や安全のための人間の行動の監視[4]など，家庭の組み込みセンサーによってキャプチャされたさまざまなデータの監視と視覚化から，エネルギー消費の最適化と改善のためのさまざまな家電製品の制御までに及んでいる．[5]，[6].

高齢者の人口が増加するにつれて，在宅ロボットと IoT によるヘルスケアおよび在宅高齢者への支援が注目を集めている[7]．家庭内のさまざまな種類の IoT デバイスは，クラウドサービスと通信して役割を達成する．例としてクラウドロボットと呼ばれるクラウドサービスとして提供される機能で動作するロボット[8]，各個人の健康状態を監視するウェアラブルデバイス[9]，多くのデバイスが検知したデータを使用したアクティビティ認識[10]が挙げられる．その結果，より多くの IoT デバイスとサービスが家庭に導入されると，家庭とさまざまなクラウドサービスの間でより多くのトラフィックが送信される．しかしながら，家庭からのインターネット接続の品質は，家庭で同時に使用されるさまざまなサービスをサポートするには不十分な場合がある．家庭での典型的なダウンロードスループットは数十 Mbps であると報告されている[11]，これは家庭で使用される多くのアプリケーションにとっては十分であるかもしれないが，高齢者支援サービスの場合は動画像データをクラウドに送出することもあるため，アップロードスループットも重要となる．ロボットの制御などではレイテンシが小さいことも求められる．さらに，一時的な輻輳や無線状態の悪さにより，インターネットやネットワーク内のスループットが低下する場合がある．限られたネットワークリソースの下で多くの IoT デバイスを用いるサービスの質を維持するには，人々の生活に大きな影響を与えるサービスがリソースの不足の影響を受けないように，ネットワークで QoS（Quality of Service）を制御することが重要になる．

著者らは，欧州の研究者と共同で，高度な ICT ロボティクスを用いた高齢者支援サービスのアジャイル型共創開発手法を定義し，３つのサービス（Mobility, Daily Life, Socialization）とサービスを支えるプラットフォームの開発を通してその研究開発手法の効果を実証することを目的に，ACCRA （Agile CoCreation of Robots for Ageing）を実施している．本稿では，ACCRA におけるネットワークの QoS 制御について述べる．

2. Qos 制御システムの概要

住居や高齢者支援施設に居住する高齢者等が，そこに設置されているロボットや複数センサー，および身につけているウェアラブルデバイス等を利用して，インターネット越

1. 京都大学学術情報メディアセンター
A) okabe@media.kyoto-u.ac.jp

Vol.2020-GN-109 No.7
Vol.2020-CDS-27 No.7
Vol.2020-DCC-24 No.7
2020/1/23

しに提供される複数の高齢者支援クラウドサービスを利用している状況を想定している．各クラウドサービスは複数のデバイスと通信し，また各デバイスも複数のクラウドサービスと通信している.本ネットワーク制御システムでは，サービスにおける各デバイスやデータの優先度や，高齢者等の生活における各サービスの優先度を利用してネットワークの制御を行う．本ネットワーク制御システムは，図 6 で示すように，以下の 3 つの入力を REST API により外部から受け取り，サービス-デバイス/データ間の通信の優先度を計算し，優先度とデータの種類に応じて，通信の可否の判断や帯域制御, Queueing の制御等を行うことを想定している．デバイス向けインターフェイス: 各デバイスが送受信するデータをネットワーク制御システムに入力する．サービス向けインターフェイス: 各サービスが利用するデバイス/データと，各デバイス/データの優先度 (デバイス/データがサービスの提供においてどの程度重要であるか) をネットワーク制御システムに入力する．高齢者・介護者向けインターフェイス: 高齢者・介護者や施設にとって各サービスがどの程度重要であるかと，各高齢者・介護者が利用しているデバイスの識別子をネットワーク制御システムに入力する．ネットワーク制御システムにおける制御の流れは図 7 のように構成されている．デバイス向け API から入力された各デバイスの情報，サービス向け API から入力された各サービスの情報，高齢者・介護者向け API から入力された各高齢者・介護者に関する情報は，ネットワーク制御計算エンジンに渡され，ネットワーク制御計算エンジンによって優先すべき通信のリストが生成される．ネットワーク制御実行エンジンはネットワーク制御計算エンジンが生成したリストに基づき，各家庭や高齢者支援施設のネットワーク機器の制御等を行う．

3. 設計方針

本研究では，家庭において高齢者を支援するロボットと多くのセンサーが，無線 LAN や有線 LAN で構成されるホームネットワークにつながり，ゲートウェイを介してクラウド上のサービスとやりとりする状況を想定し, ロボット，センサー，ゲートウェイ，ならびにクラウド上のサービス間の通信のフロー単位での QoS 制御を行うことを考える．対象アプリケーションとしては、動画ストリーミング、音声などのリアルタイムストリーミング、レイテンシセンシティブなものを含むセンサデータやロボット制御データ、音声対話のためのテキスト情報など多様なものを扱う。使用されるプロトコルとしては FIWARE を介した pub/sub（RESTful HTTP）、MQTT、RTMP などを想定するが、特に限定はせず任意の TCP および UDP のフローが扱えるようにする。

IP ネットワークにおけるフロー単位の QoS 制御のための従来技術として IntServ [12]や DiffServ [13]が標準化されているが，広く使われるには至っていない．本研究では，OpenFlow に代表される Software Defined Networking [14]を前提とし，SDN コントローラによる一元的な管理で対象となるネットワークの QoS 制御を行うアーキテクチャを採用する．アプリケーションからの QoS 要求を受け付けるアドミッション制御には，IntServ における RSVP のような専用のプロトコルは用いず，単純な REST API による HTTP 通信で実装するものとする．

多くのホームネットワークで，IoT デバイス，サービスプロバイダー，およびユーザ（高齢者・介護者）からなるシステム全体とそれぞれのコンテキストを把握している管理者が不在である．その制約を前提に，SDN コントローラとして動作する Controller は，入力として，デバイス用，サービス用，ユーザ用の 3 種類の API を提供する．Controller は，入力として登録された情報を集約し，各フローのグローバル優先度を自動的に推定し，グローバル優先度とネットワーク状態に従ってワイヤレスアクセスポイント，スイッチ，CPE（Customer Premises Equipment）などのネットワークデバイスを構成する．

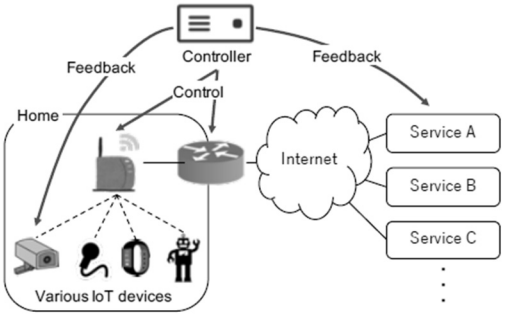

図 1　提案するアーキテクチャの環境とユースケース

4. QoS 制御システムのアーキテクチャ

図 2 に, Controller の提案されたアーキテクチャの概要を示す.

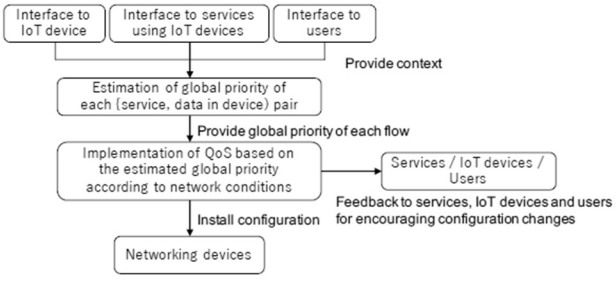

図 2　アーキテクチャの概要

4.1 IoT デバイス/サービス/ユーザーへの API

最初のステップは, Controller が IoT デバイス, サービス, およびユーザーに関連するコンテキスト情報を取得することである．入力を受信するために，Controller には各パーティ（IoT デバイス，サービス，ユーザー）に適した 3 種類

Vol.2020-GN-109 No.7
Vol.2020-CDS-27 No.7
Vol.2020-DCC-24 No.7
2020/1/23

の Interface がある．各 Interface は，次の原則に従って設計されている．IoT デバイスへの Interface は，デバイスが送受信できるデータの種類を取得することである．この目的は，FIWARE [15]や WoT [16]などの既存の IoT プラットフォームのデバイスの抽象化に非常に似ており，(プロパティ/属性，その値のタイプ) ペアのセットとして表されるため，FIWARE または WoT への入力を可能な限り再利用できるように，それらと同じモデル構成を持っており，この情報はサービスとユーザーに依存しないため，デバイスはこの Interface を介して自動的に情報を提供するものと想定される．プロパティ/属性とそのタイプのリストに加え，各プロパティ/属性について，Controller はデータを提供および処理するためにデバイスが通信するクラウドサービスを知る必要がある（IoT のコンテキストでは Coflow [22] QoS を適用するという目的は，このような IoT Coflow を考慮しないと達成できないためである．）サービスへの Interface に要求されるのは，デバイスがサービスにアクセスするデータの種類を取得することである．サービスは，デバイスと通信するサービスのエンドポイント (IP アドレス, URL など) とともに，ユーザーが所有するデバイスにプロパティ/属性のリストを登録する．セクション III-A で説明したように，データの重要性はサービスに依存する場合がある．つまり，サービスを提供するために他のデータよりも重要でないデータもある．そのため，サービスは各プロパティ/属性の優先度などの重要性も提供する必要がある．そのため，ユーザーはこの Interface を介して（サービス，生活における優先順位）のリストを提供することが期待されている．

4.2 各（サービス，デバイス内のデータ）ペアのグローバル優先順位の推定

2 番目のステップは，Interface からの 3 種類の入力から，ユーザーの寿命に対する各（サービス，デバイス内のデータ）ペアのグローバル優先順位を推定することである．グローバル優先度は，家に住む人々が使用するすべての IoT デバイスとサービスを統合するという観点から割り当てられた優先度である．この推定の基本方針は，ユーザーが重要だと考えるサービス，およびユーザーに良質のサービスを提供するために重要な (サービス, デバイス内のデータ) のペアを優先することである．推定の 2 つの戦略の間にはトレードオフがある． 1 つは，ユーザーが同時に提供できるサービスの数を最大化することである．これを実現するために，サービスによって提供される高い優先度を持つ (サービス，デバイス内のデータ) ペアは，高いグローバル優先度を持つ．もう 1 つは，ユーザーに同時に提供できる各サービスの品質を最大化することである．これは，重要なサービスのペア（サービス，デバイス内のデータ）に高い優先度を割り当てることで実現できる．

4.3 推定グローバル優先度に基づく QoS 設定

3 番目のステップは，ネットワーク状態を考慮して，前のステップで推定されたグローバルプライオリティをネットワークトラフィックに適用することである．このステップでは，フローを優先度順にネットワークに詰め込む単純な貪欲なアルゴリズムを利用する．構成が決定したら，Controller はネットワークデバイスを構成して，各フローに QoS を適用する．実際の動作は以下のようになる。

① アプリケーションから帯域(bps)/遅延(ms)/破棄レベル(3 段階)を指定してフローを作成/変更/削除できるようにする REST API を設計・実装する。

② 各 API において、帯域/遅延/パケット破棄レベルの要求を満たせないため拒否した場合にそれを通知するためのエラーコードを返すとともに、その際、可能な帯域/遅延/パケット破棄レベルも返す。

③ 新しいフローを要求された場合、それまでに使用されているフローも含めて改めて優先度順に割り当て直す。

④ 新しいフローの方が優先度の高い場合、既存の低優先度フローはベストエフォートにする。

このステップのもう 1 つの役割は，優先度推定に基づくフローの QoS 制御の可否をデバイスとサービスにフィードバックすることである．デバイスおよびサービスでのこのフィードバックの使用はオプションであるが，可能であればユーザーに提供される各サービスの品質を向上させるために，このフィードバックを使用してデバイス間で交換されるデータの品質と優先度を変更することが期待される．たとえば，監視サービスでは，Controller からのネットワークリソースが不十分なときは，可能な限りサービスの提供を継続するため，カメラで撮影した映像の伝送を停止し，マイクでキャプチャした音声のみを伝送する．サービスとデバイスの動作が変更されると，Controller の対応する Interface を介して新しい情報が提供され，Controller は新しいグローバル優先順位でネットワーク構成を更新する．

4.4 ソフトステートによる動作

ACCRA におけるネットワーク制御システムの背景を図 3 に示す．

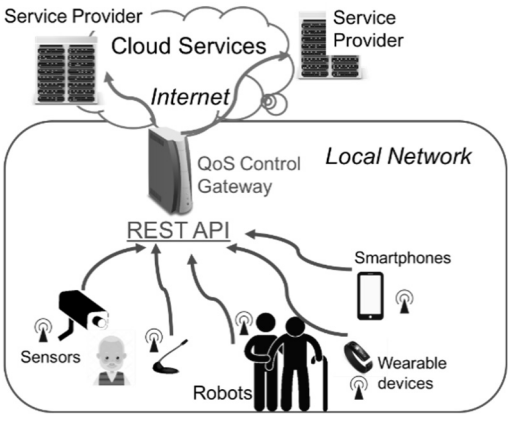

図 3　ACCRA におけるネットワーク制御システム背景

Vol.2020-GN-109 No.7
Vol.2020-CDS-27 No.7
Vol.2020-DCC-24 No.7
2020/1/23

ACCRA ネットワーク制御システムでは，優先度に基づくフローの許可/拒否は，デバイス向け API のデバイス（プロパティ）情報登録/更新，高齢者・介護者向け API の利用者（デバイス）情報登録/更新，サービス向け API のサービス利用者デバイス情報登録/更新のいずれかのタイミングで行われる．基本的には「デバイス付帯送受信情報更新 :update」で送受信の開始が要求され，計算された優先度順にフローが許可され当該フローに関して事前に設定された OpenFlow スイッチの 2 つのポート間の通信を，フローエントリを設定することで制御される．もしフローが許可されている状態で，「利用者サービス情報更新 :update」によりサービスの優先度が小さい値に変更された場合には，新しい優先度に基づいて許可/拒否が再計算され，設定済みのフローが切断されることがあるという動作である．その場合は，フローを送信している側に切断の理由を直接エラー通知する術はなく，送信中のパケットに対する QoS 保証がなくなり廃棄が起こりうるようになる．すなわち送信側は送信できているつもりでいるのに受信側は受け取れていない状態が半永久的に続く可能性が否定できない．

また，「デバイス付帯送受信情報更新 :update」で設定されたフローは，「デバイス付帯送受信情報削除 :delete」でフローが削除されるか，あるいは「デバイス基本情報削除 :delete」あるいは「デバイスデータ詳細情報削除 :delete」でそのフローの前提となっている情報が API により明示的に削除されない限り永続的 に残る．例えば送信側のデバイスが突然停止してしまったような場合には，そのままではフローは削除されない．

これらの問題を合わせて解決するために，定期的に生存確認を行いタイムアウトが行えるようにするソフトステート(soft state)の機能を導入する．動作は具体的には以下の通りとする．システムのタイムアウト値として次の二つを定義する「**フロータイムアウト値**: 自然数，単位は秒，default 値は 300,」「**デバイスタイムアウト値**: 自然数，単位は秒，default 値は 3,600,」さらにタイムアウト値が 0 または負数のときは，タイムアウトは行わない（ハードステート）．デバイスタイムアウト値が正の値の場合，デバイス基本データ情報ないしデバイスデータ詳細情報が登録ないし更新されて以後デバイスタイムアウト値で示される秒数以内にデバイス基本データ情報の更新，あるいはデバイスデータ詳細情報ないしデバイス付帯送受信情報の登録・更新・削除のいずれの操作も行われなかった場合，当該デバイス基本データならびにデバイスデータ詳細情報を削除し，関連するフローを削除する．デバイスのタイムアウト後に当該デバイスに関してデバイス基本データ情報の更新・削除，デバイスデータ詳細情報ないしデバイス付帯送受信情報の登録・更新・削除が行われた場合は，エラーコードによりエラーを返してデバイス側が識別できるようにする．

5. QoS 制御システムの動作

QoS 制御システムの動作を検証するために，二つのシナリオを用意し，これを実演できる環境を整えた．

一つ目のシナリオ（以下，シナリオ 1）は，内部ネットワークにあるサーバーを経由して，同ネットワーク内の二つのクライアント同士がメッセージのやり取りを行うものである．

二つ目のシナリオ（以下，シナリオ 2）は，内部ネットワーク内のクライアントが，インターネット上のサーバーを経由して，他ネットワーク内のクライアントとメッセージ及びリアルタイム映像のやり取りを行うものである．シナリオ 2 においては，内部ネットワークから外部ネットワーク（さらには，インターネット）に接続する必要があるため，ネットワーク境界に iptables を使用した NAPT 型 Gateway を設置した．以上いずれのシナリオにおいても，Openflow SW を ACCRA controller で管理することでフロー制御を行っている．

また，クライアントには smartphone 又はノートパソコンその他の情報機器を使用する．内部ネットワーク内には DHCP サーバーがないため，これらの機器には手動でネットワーク関連設定を行う必要がある．

また，ACCRA controller は REST API を有しているが，デモにおける利便性向上のため，GUI 操作で ACCRA controller を設定できるユーティリティ（REST GUI server）を用意した．このユーティリティは ACCRA controller と同じ機器にホストされた Web アプリケーションである．使用するためには，ブラウザを立ち上げ，http://<ip addr>/accra と URL 欄に入力してエンターキーを押す．ここで，<ip addr>とは ACCRA controller に割り当てられた ip アドレスのことである．必要な機材及びその名称について，図 4 及び図 5 に示す．

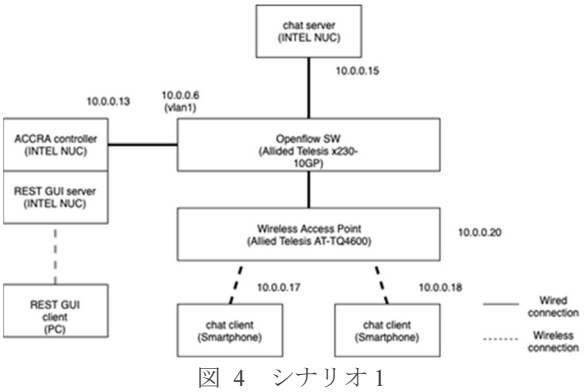

図 4　シナリオ 1

Vol.2020-GN-109 No.7
Vol.2020-CDS-27 No.7
Vol.2020-DCC-24 No.7
2020/1/23

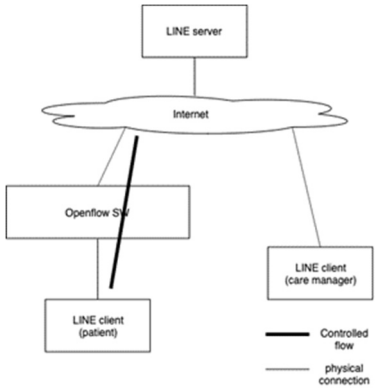

図 5　シナリオ 2

以下では，同資料の記述に基づき説明する．内部ネットワーク内の機器には全て IP アドレスが割り当てられており，そのネットワークアドレスは 10.0.0.0/24 となっている．また，外部ネットワーク（青線で示した）に接続する際の IP アドレスには特筆すべき条件はない．すなわち，DHCP での自動設定を用いても良い．内部ネットワークに接続するクライアント用情報機器には，手動でネットワーク関連設定を行う必要があるのは既に述べた．ここでは，具体的な手順の一例を紹介する．

IP アドレスは別資料（シナリオ詳細）に記載されているものを用いる．その他に，Gaateway として 10.0.0.15，ネットワークマスクとして 255.255.255.0(24)を設定する．さらに，DNS サーバーとして 8.8.8.8(Google DNS)を設定する．DNS サーバーを手動で設定した場合，一部の公共 Wifi（KUINS-AIR など）は利用できなくなる場合があるので注意すること．起動に成功したのち，各種サービスの立ち上げ及び設定を行う．まず，Gateway において iptables の設定を行う．そのために，以下を Gateway で実行する．$ sudo iptables-restore < /etc/iptables.rule 次に，ACCRA controller を起動するため，以下を ACCRA controller で実行する．$ cd ~/accra$ ./accra_start.sh これは，ACCRA ネットワーク制御システム マニュアル 20190823.md に記述されている処理である．なお，Openflow スイッチは起動時に Controller との接続を試みるので，Openflow スイッチを起動する前に ACCRA controller を起動し，スイッチに接続すること．Openflow スイッチにサーバー及びクライアントを接続する際には，ポート番号に注意する．1-4 ポートは通常のスイッチポート，5-8 ポートが Openflow ポートとなっており，Controller は通常ポート，制御対象は Openflow ポートに接続する．システムが正常に動作しているかどうかを確認，おおよそ以下の項目を確かめればよい．

まずは，Gateway 接続を確認する．Gateway を通常ポートに繋ぎ変え，ACCRA controller で$ curl www.google.com を実行し，成功すれば Gateway 接続は正常に動作している．次に，ACCRA controller がスイッチを認識しているか確かめる．これには，ACCRAControlle 上で，$ curl localhost:8080/firewall/log/status を実行し，応答中に switchId が含まれているかを確認する．

Gateway 接続が出来なかった場合，各機器での IP ルーティングが適切ではないことが考えられる．IP ルーティングテーブルを確認するには，$ ip route show を実行する．この結果がおおよそ以下のようであれば正常である．- ACCRA controller においては default via 10.0.0.15 dev eno1 10.0.0.0/24 dev eno1<外部ネットワークアドレス> dev wlp58s0- Gateway においては default via <外部ネットワークの Gateway アドレス> dev wlp58s010.0.0.0/24 dev eno1 結果がこれと異なる場合は，ip route add コマンド，もしくは ip route del コマンドを用いて調整する．

諸注意として､外部ネットワークと内部ネットワークはネットワークアドレスにより完全に区別できなければならない．これはすなわち，外部ネットワークのネットワークアドレスは 10.0.0.0/24 を含んではならないことを意味する．区別が出来ない場合，Gateway において外部ネットワーク向けのパケットが内部ネットワークに送信され，タイムアウトとなる可能性がある．ネットワークの疎通確認には ping がよく使われるが，NAPT Gateway は ping を通さないため，正常であったとしても ping に対する応答は返ってこない．代わりに，dig や curl など icmp を使用しない手法を用いること．公共 wifi では，接続機器同士の通信を禁止している場合がよくある（KUINS-AIR も該当する）．そのため，REST GUI server と RESET GUI client 間の通信が出来ないことがある．この場合には，REST GUI client に適切なネットワーク設定を行ったのち，Openflow スイッチの通常ポートに接続し，http://10.0.0.13/accra で RESTGUI server に接続する．適切なネットワーク設定の一例としては，IP アドレス 10.0.0.30（任意の未使用 IP アドレス），ネットワークマスク 255.255.255.0（固定），ゲートウェイ 10.0.0.15（固定）とする．

6. まとめ

本研究は，多くの IoT デバイスとサービスを備えたホームネットワークで QoS を管理する Controller のアーキテクチャを提案し，ユーザーが多くのサービスを快適に使用できるようにすることが目的である．重要な機能の 1 つは，提案されたアーキテクチャは，多くの家庭で現実的となるネットワークおよびシステム全体を十分に理解している管理者がいないという前提の下で設計されていることである．IoT デバイス，サービス，およびユーザーの 3 つの関係者の入力からのグローバル優先順位の自動推定は，ホームネットワークの QoS に必要なポリシーを導き出すのに役立つものである．実際のサービス開発の観点からの Interface の広範な評価，およびグローバルな優先度の推定と洗練された QoS 実装アルゴリズムの統合のためのより洗練されたアルゴリズムの設計が含まれる．

Vol.2020-GN-109 No.7
Vol.2020-CDS-27 No.7
Vol.2020-DCC-24 No.7
2020/1/23

謝辞 日頃から議論下さる ACCRA プロジェクトのメンバーに感謝する．QoS 制御システムの実装にあたって貴重なコメントと支援を提供してくれた SRC Software の森岡均氏に深謝する．なお本研究は，情報通信研究機構の委託研究として実施されている．

参考文献

1) J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things(iot): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1645 – 1660, 2013.
2) J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things(iot): A vision, architectural elements, and future directions,” FutureGeneration Computer Systems, vol. 29, no. 7, pp. 1645 – 1660, 2013.
3) D.-M. Han and J.-H. Lim, “Smart home energy management system using IEEE 802.15. 4 and zigbee,” IEEE Transactions on Consumer Electronics, vol. 56, no. 3, pp. 1403–1410, 2010.
4) N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, “Monitoring behavior in home using a smart fall sensor and position sensors,” in 1st Annual International IEEEEMBS Special Topic Conference on Microtechnologies in Medicine and Biology. Proceedings (Cat. No.00EX451), 2000, pp. 607–610.
5) J. Choi, D. Shin, and D. Shin, “Research and implementation of the context-aware middleware for controlling home appliances,” IEEE Transactions on Consumer Electronics, vol. 51, no. 1, pp. 301–306, Feb 2005.
6) K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson, “Scheduling smart home appliances using mixed integer linear programming,” in 2011 50th IEEE Conference on Decision and Control and European Control Conference, Dec 2011, pp. 5144–5149.
7) K. Yamazaki, R. Ueda, S. Nozawa, M. Kojima, K. Okada, K. Matsumoto, M. Ishikawa, I. Shimoyama, and M. Inaba, “Home-Assistant Robot for an Aging Society,” Proceedings of the IEEE, vol. 100, no. 8, pp. 2429–2441, Aug 2012.
8) J. Kuffner, “Cloud-Enabled Robotics,” in 2010 IEEE-RAS International Conference on Humanoid Robotics, Dec 2010.
9) M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, and S. Andreescu, “Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges,” in IEEE SCC 2015, June 2015, pp. 285–292.
10) F. Cicirelli, G. Fortino, A. Giordano, A. Guerrieri, G. Spezzano, and A. Vinci, “On the Design of Smart Homes: A Framework for Activity Recognition in Home Environment,” Journal of Medical Systems,vol. 40, no. 9, p. 200, Jul 2016.
11) M. Isla, “The World’s Internet Speeds Increased More than 30You Keeping Up?” http://www.speedtest.net/insights/blog/global-speed-2017/, accessed: 2018-07-31.
12) R. Braden, D. Clark, and S. Shenker, “Integrated Services in the Internet Architecture: an Overview,” June 1994, RFC1633. [Online]. Available: http://tools.ietf.org/rfc/rfc1633.txt.
13) S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services,” December 1998.
14) M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “PolicyCop:An Autonomic QoS Policy Enforcement Framework for Software Defined Networks,” in IEEE SDN4FNS 2013, Nov 2013, pp. 1–7.
15) “FIWARE,” https://www.fiware.org/.
16) W. W. W. Consortium, “Web of Things at W3C,” https://www.w3.org/WoT/.