

Pub/Sub 機構の高度化に関する研究

本間 可楠¹ 知念 賢一¹ 篠田 陽一²

概要：センサやデバイス、ネットワーク技術の発展に伴い、社会のあらゆる場所に情報通信システムが展開され、様々な情報が交換・収集・解析されている。こういった情報通信システム間で扱われる通信モデルでは、システム間連携における依存度を低減させるため、メッセージの送信者が特定の受信者を想定せずにメッセージを送信する非同期メッセージモデルである Publish/Subscribe モデル (以下、Pub/Sub モデル) を利用する。このような特徴を保持したまま映像や音声のような同期性や等時性のあるデータを配送したい場合がある。しかし、非同期通信という特性上、同期性や等時性を有するようなデータ配送が困難であるという欠点が存在する。本研究では、Pub/Sub モデルにおける通信形態や配送するデータの意味を保持しつつも、同期性や等時性というデータ配送を可能にし、データの種類ごとに適した通信の確立とブローカにおける膨大なデータへの対応を目的に、提案システムに求められる高度化のための配送手法および提案システムの構成を論じる。

Research on Advanced Pub/Sub Mechanism

Kanan HOMMA¹ Ken-ichi CHINEN¹ Yoichi SHINODA²

1. はじめに

Internet of Things (IoT) や Cyber-Physical System (CPS) による高度に情報化された社会の実現に向けて、多様なシステムやサービスを構築・連携可能な基盤が求められている。こういった連携を容易に実現するためには、送受信者がお互いのネットワーク上における所在 (IP アドレス等) を明確に指定する位置指向の通信ではなく、目的やデータの内容に応じたコンテンツ指向の通信 [1] を扱えることが望ましい。こうしたコンテンツ指向の概念を支えるメッセージパラダイムの一つに、Publish/Subscribe モデル (以下、Pub/Sub モデル) がある。

この通信モデルは、システム間連携における依存度を低減させるために、メッセージの送信者が特定の受信者を想定せずにメッセージを送信する非同期メッセージングモデルとなっている。このような形態から、Pub と Sub は互いに相手を知っている必要はなく、メッセージを受信す

る場合にはトピックを事前に指定しておくだけでよい。また、メッセージを送信する場合にはトピックを指定し公開すればよく、事前に受信登録された相手すべてに動的に配信される [2]。

前述の特性から Pub/Sub モデルは非力なデバイスや送受信関係が流動的な状況に適する。このため、近年では特に IoT への応用が活発で、IoT 向けの各種プラットフォームにおいて Message Queuing Telemetry Transport (MQTT) プロトコル [3] 等の適用がなされている [4]。

しかし、このような Pub/Sub モデルを用いた機構では、非同期通信という特性上、映像や音声のような同期性、等時性があるようなデータ配送が困難であるという欠点が存在する。また、配送するデータ量が大きければその分帯域に対する処理も必要になってくる。一般的に非同期性という特徴を有したまま等時性や同期性を保有したデータを配送したい場合、データの種類に応じた複数のプロトコルを組み合わせた配送が行える。しかし、これは設計時に想定された機能やデータ構造を前提としているのであって、仕様外の独自の機能を加えた複数のプロトコルやシステムを扱うことによる不必要なリソースの浪費が発生しえる。また、環境に応じて異なる実装を施したい場合には、結果的

¹ 北陸先端科学技術大学院大学先端科学技術研究科
Graduate School of Advanced Science and Technology,
Japan Advanced Institute of Science and Technology

² 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,
Japan Advanced Institute of Science and Technology

に密結合な通信形態となり依存性が高まることに繋がり、スケーラビリティの確保が難しくなる。本研究では、同期性や等時性、非同期性を持ったメッセージ配送に着目し、Pub/Sub 機構の高度化によって、データの種類ごとに適した通信の確立とブローカに対する膨大なデータへの対応を実現する。これにより、Pub/Sub 機構における通信形態や配送するデータが有している意味を保持しつつも、同期性や等時性という特徴を持ったデータの配送を可能にする。

2. 基盤技術および関連研究

2.1 MQTT

MQTT は M2M 接続の Pub/Sub メッセージングモデルのプロトコルであり、サーバからデータの通信を開始するプッシュ型プロトコルである [3]。MQTT は、遅延が大きく狭帯域のネットワークにおいて、正確なデータ送信を行うことを目的としている。MQTT を用いる利点としては、接続の一時的な切断に強い点や帯域の利用効率が良く軽量である点が挙げられ、信頼性や通信速度の低いネットワークでも利用に有効である。

一般的な MQTT はメッセージを送信する Publisher とそれを受信する Subscriber、さらに、それらを仲介する Broker の 3 つの要素で構成される (図 1)。MQTT では、トピックと呼ばれるメッセージに付属する件名を表す文字列を用いて Subscriber が受信したいメッセージを Broker に伝達する。ここで、トピックは 1 つのメッセージに対して複数指定することが可能である。Broker は Publisher からメッセージを受信した後、トピックの情報からメッセージを要求する Subscriber へ送信を行う。送信先の存在しないメッセージに関してはトピックごとに最新のメッセージのみ保持することが可能であり、新たに送信先となる Subscriber からの接続があった際に送信される。

MQTT のメッセージフォーマットを図 2 に示す。MQTT のメッセージは、全てのメッセージに存在する制御用の固定長ヘッダ、いくつかのメッセージに存在する可変長ヘッダと可変長のメッセージペイロードで構成されている。

→ data flow

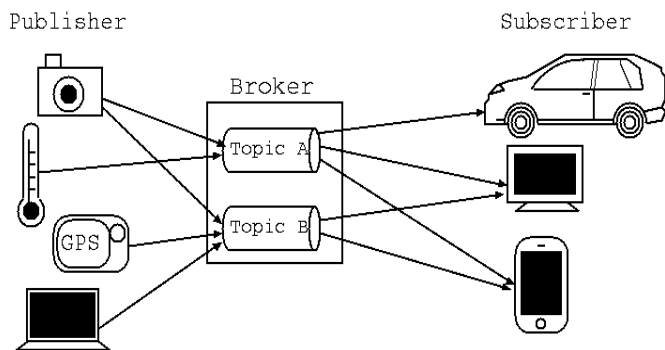


図 1 MQTT のアーキテクチャ

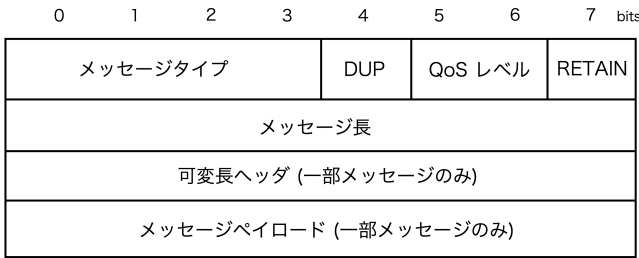


図 2 MQTT メッセージフォーマット

固定長ヘッダはメッセージのタイプを表すフィールドと、QoS レベルを含めた各種フラグ、さらに残りのメッセージ長を示すフィールドが含まれている。固定長ヘッダの大きさは 2 bytes と軽量でありオーバーヘッドは小さい。また、3 段階の QoS レベルに基づいた到達保証が可能でありメッセージ送信の正確さも備えている。

しかし、これらの特徴を保持したまま、同期性や等時性をもったメッセージ送信の機能は保証されない。そのため、映像や動画などのコンテンツを配信したいとなった場合には、また別のプロトコルを採用するといった手間が発生し、管理の範囲も拡大する。そのため、MQTT においても、同期性や等時性の特徴を有したメッセージ配送を検討する必要がある。

2.2 配送するデータに関わる手法

非同期通信の特徴を有する Pub/Sub モデルにおいて、データ配送に関する既存手法には様々な手法が提案されている。

坂野ら [5] はオーバーレイネットワーク上の配送経路長を可能な限り抑えるようなストリームデータの配送を目的に、ストリームデータ配送に適した分散 Pub/Sub の提案をしている。映像などのストリームデータを想定した場合、ネットワーク資源の消費が激しいことが予想される。一般的な Pub/Sub では、アプリケーションが提供するコンテンツの要求と、実ネットワークにおける位置を把握した上での通信との対応づけをサーバが集中的に担うが、その場合、サーバに負荷が集中する問題点がある。これを解決するために、Pub/Sub を非集中的に実現する分散 Pub/Sub がある [6]。ここでは、このような手法に対する懸念および検証における評価指標として 3 点が挙げられている。

- (1) 配送経路長
- (2) 負荷分散の公平性
- (3) Publish の停止性

これらの懸念を解消するために、構造化オーバーレイネットワークの技術を用いたストリームデータの配送に適した分散 Pub/Sub を実装および実験し、既存手法との比較を

行っている。結果では、配送経路長については、経路長がトピックに登録されている Pub と Sub の個数には影響を受けていないのに対し、既存手法ではトピックへの登録数に関わらず経路長が長くなっていることが示されている。負荷分散の公平性については、トピックに登録してある各ノードにおけるメッセージの送信回数、受信回数、転送回数から、それぞれに対する相関関係を算出し、Pub に関するメッセージ転送の負荷が Pub と Sub 間でどのように分散しているかの確認を行った結果、転送回数が比較的少なく抑えられていることもあり、負荷分散の不公平性が生じていないとしている。Publish の停止性については、トピックに Sub が存在しない場合の Publish の停止性を確かめた結果、Sub が 0 になった期間はメッセージ数が 0 になっており、ストリームデータのようなサイズの大きなデータを相手のいないトピックに配送することはなく、ネットワーク資源の消費を抑えたことを示している。

さらに、データごとに異なる遅延耐性を考慮した研究もある。内山ら [7] は、データごとに異なる遅延耐性に着目し、IoT システムにおいて、低品質なネットワークを用いた場合でもシステムの信頼性や可用性を向上させるため、送信データの優先度に応じた送信順制御が可能な MQTT プロトコルの提案をしている。MQTT では、メッセージの優先順位は考慮されておらず、データの特性と関係なくデータ送信が行われる。そのため、通信帯域や速度の品質が低いネットワークを用いたアプリケーションでは、データの活用の際に重要度の低いデータ送信で帯域を圧迫し、重要なデータ送信の遅延や信頼性の低下が予想される。これを解決するために、コンテンツごとに定められた優先度の情報をもとに送信順制御が可能な MQTT を実装し評価を行っている。Pub から送信されたメッセージが Sub に届くまでの時間である遅延時間を評価指標とし、送信順制御が可能な MQTT を適用することで優先度が高いメッセージの遅延時間減少を確認できれば優先度に応じた制御の効果があるという前提のもと実証している。結果として、通信速度の低いネットワークにおいて高い優先度のメッセージは遅延時間が既存の MQTT に比べ 4 ～ 6 % 程度減少し、メッセージの送信時間が長くなる低速なネットワークほど送信順制御が可能な MQTT が効果的であるとしている。

しかし、これらの先行研究はストリームデータに特化した処理や、コンテンツごとに定められた優先度の情報をもとにメッセージ配送を行っているにすぎず、様々な異種要素を組み合わせ異なる種類のデータを配送したいとなった場合には、データごとに適した配送手法を適用することになり、結果的に複数のプロトコルを組み込むような仕組みが必要になる。これはスケラビリティの確保が難しくなり、軽快なメッセージ配送とは言いがたくなる。

2.3 ブローカの処理量に関する手法

Pub/Sub モデルを、多数の IoT デバイスなどから成る大規模システムへの適用を考えた場合、ブローカへの負荷が集中することによるサービス停止や QoS 低下が問題となる。この場合、複数のブローカへのスケールアウトにより負荷分散等を実現する手法が提案されている [8], [9]。これらの手法では、ブローカ群によるオーバレイネットワークを形成し、ブローカ同士の協調によるブローカの分散化を実現している。

しかし、これらの既存手法では、スケールアウトにより高いスループットが得られる一方で、分散処理のオーバーヘッドとしてメッセージの配送遅延が増大するという問題がある。ブローカへの負担を考慮しつつも、通信の形態および異なる特性を保持したデータの配送を軽快に行える仕組みが必要である。

3. 提案と実装

3.1 一般的なメッセージ配送の定義

本研究で提案する Pub/Sub 機構におけるメッセージ配送とそのデータの種類について定義する。まず、メッセージ配送で着目すべきこととして、通信の形態が永続であるか一時であるかということと、同期か非同期であるかという点である。永続通信の本質は、メッセージ転送を行うために、送信者も受信者も動作し続ける必要はない。一時通信には、記憶機能は提供されていないため、受信者は、メッセージが送られてくるときに、そのメッセージに応答する準備ができていなければいけない。非同期通信では、送信者は、メッセージが転送を受けつけられた直後、継続して動作することが可能である。同期通信では、送信者は、少なくとも 1 つのメッセージが受け取られるまでブロックされる。さらに、通信の全く異なった形式としてストリームがある。連続するメッセージは時間的な関係性を持つかどうかを考慮する必要がある。

このような通信の形態が様々な状況を想定した場合、Pub/Sub 機構における非同期通信の形態を保持したまま、同期性や等時性といった特徴を持つデータの配送を行うには次のことが起こり得ることを明確にしておく必要がある。

- 喪失 (つまり配信されない)
- 重複 (複数のコピーが到着する)
- 遅延 (データの種類によっては、到着までに長い時間がかかる)
- 順序を違えての到着

たとえばストリームデータを多対多でやり取りするサービスが展開された場合、蓄積や配信のコストが非常に大きくなる。ブローカから Sub へのデータの配送において、データの喪失は大いに考えられる。さらに Sub や Pub が増加した分、膨大なデータおよび通信によるブローカへの負荷は避けられないことを考慮すると、データの種類によ

ては遅延も発生する。このような遅延の発生は、Pub/Sub 機構本来のリアルタイム性を損ない、適用範囲を狭める。また、メッセージが喪失したことによるメッセージの再配送が生じた場合、もしくは受信者がチャンネルへ不在である場合に、Pub からメッセージ配送がされ続けるために、メッセージが重複して配送されることも十分考えられる。その結果、ブローカにはデータが蓄積され続ける。

これらは、既存手法によって解消されるが、特定の機能に特化したものがほとんどであり、非同期通信の形態を保持したまま、異種要素の特徴を有したデータの配送をしたい場合には、複数のプロトコルを組み合わせる必要がありスケーラビリティの保証を確保できないといった懸念がある。

本稿では単に特定の機能に特化した Pub/Sub 機構の拡張を行うのではなく、通信の形態やデータの種類の応じたメッセージ配送の確立を高度化とし、新たに機構として Pub/Sub 通信に適用するものである。

3.2 高度化のための配送手法

前章で整理した内容をもとに、本研究で提案するシステムに必要となるメッセージ配送手法を高度化とし、これらの手法について検討する (表 1)。

3.2.1 タイムスタンプ

一般の非同期な通信環境には遅延時間のゆらぎが存在し、このために映像や音声といったストリームデータを配送する際に時間軸が乱れてしまうという問題がある。

これを解決するために、基準クロックをカウントするカウンタによって、配送する各メッセージにタイムスタンプを付加する。これにより、ストリームデータの時間軸情報を現実の時間経過ではなく、データとして処理することを可能にする。

たとえば、図 3 に示すように、ストリームデータにおける時間軸 t に対し、サイズに応じて分割した各データにタイムスタンプを付与する。データ単位の不規則なストリームデータであってもタイムスタンプを参照し、それに応じたデータを受け取ることが可能である。

連続的なストリームデータにとってはタイミングが非常に重要であり、タイムスタンプの処理により等時性の特徴を保持したデータ配送を実現する。また、映像や音声のようなメディアでは、送信途中に喪失した古いデータは意味を成さなくなる。この場合、タイムスタンプに応じたデータの削除が可能となるため、不必要となったデータの扱いも容易となる。

表 1 高度化のための配送手法	
手法	効果
タイムスタンプ	不必要なデータを削除可能
ポインタ	ブローカの扱うデータ数が減る

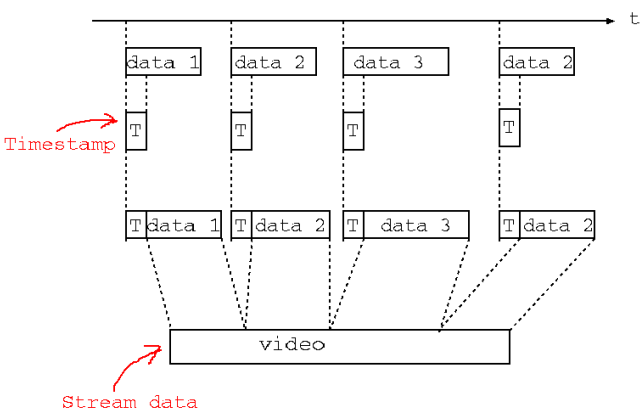


図 3 タイムスタンプを用いた構成

3.2.2 ポインタ

通常の Pub/Sub 機構におけるブローカは、Pub から一方的にメッセージを受信するため、たとえば Pub が n 個多くなれば、その分だけブローカが受け取るメッセージ量も n 倍増加し、処理負担が大きくなる。

提案するシステムでは、ブローカにはデータ本体と紐づいたポインタを流し、データ本体は別のストレージに格納されるような形態を提案する。本来、一般的なブローカではデータ本体を扱う。このため、膨大なデータを扱う場合には、ブローカのキューが溢れやすく軽快なメッセージ配送も難しくなるといった懸念がある。データのサイズに留意すると、現在では、画面解像度がフル HD (1920x1080) から 4K (3840x2160) に対応したスマートフォンやタブレットなどの携帯端末が一般的になっており、それに伴って撮影される画像はおおむね 4000x3000 画素でファイルサイズは 5 MB から 10 MB 程度になる。一方、ポインタのサイズは、たとえば URL であれば数 100 バイトで十分である。文献 [10] では 144 バイトで十分とされている。これらのことを考慮した上で膨大なデータを扱う際に、データ本体を処理する場合に比べ、データを示すポインタを処理する場合の方が高速になる。ブローカがポインタのみを扱うようになれば、ブローカ自体はより高速に動作し、Pub や Sub が増加したとしても溢れることなく軽快な処理が期待される。

さらに、ポインタを用いることにより、ブローカやチャンネルが複数あった場合には、ブローカ間でのメッセージのリダイレクト処理を容易に実現可能となる (図 4)。これは Sub が不在となった場合や、ブローカ間の連携を行いたい場合に有効で、自律分散的にメッセージのやりとりを行うことが可能である。

3.3 提案システムの設計

提案手法では、Pub/Sub 機構においてデータ本体そのものとは別のサーバで管理するという形態や時間粒度の変化に対応したモデルを新たに定義する。これにより得られ

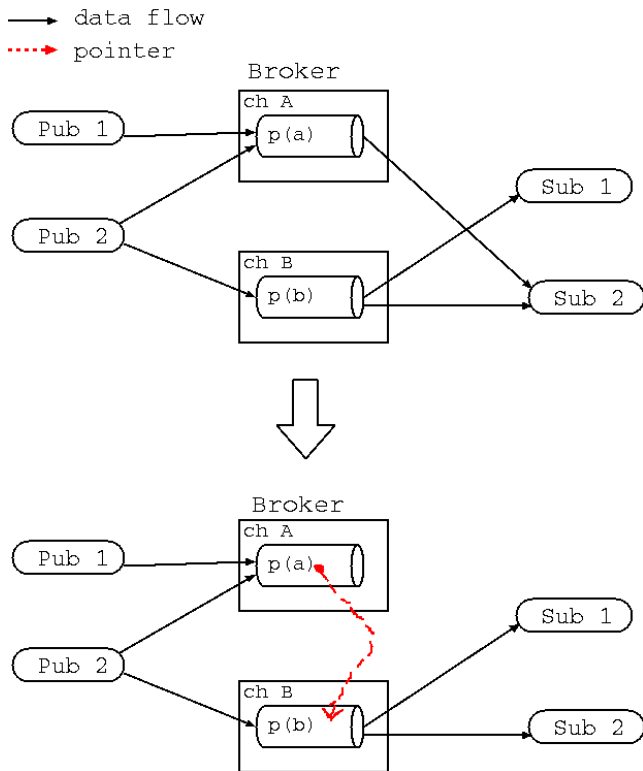


図 4 ポインタを用いた構成

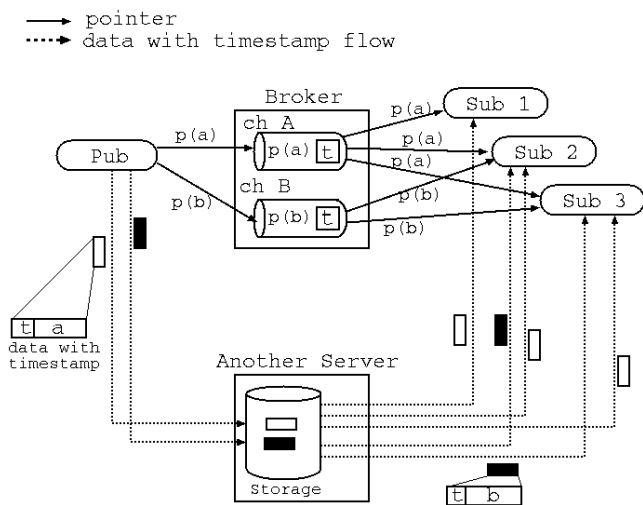


図 5 Pub/Sub 機構の高度化の実装概念

る効果を前提とした、高度化を実現する Pub/Sub 機構のシステムの構成を考える。

前述で示したタイムスタンプとポインタを例に用いた実装概念を図 5 に示す。ブローカには 2 つのチャンネルが存在し、各チャンネルにはデータの場所を示すポインタを、別のサーバにはポインタと紐づいているタイムスタンプ付きのデータ本体が格納される。音声や映像のデータでは、古くなったデータはほとんど意味を成さないためタイムスタンプに対応した不必要なデータは配送せず削除可能となる。結果的に、ブローカはデータ量の処理を抑えることができ、より高速な動作を可能とする。

→ pointer
 data with timestamp flow

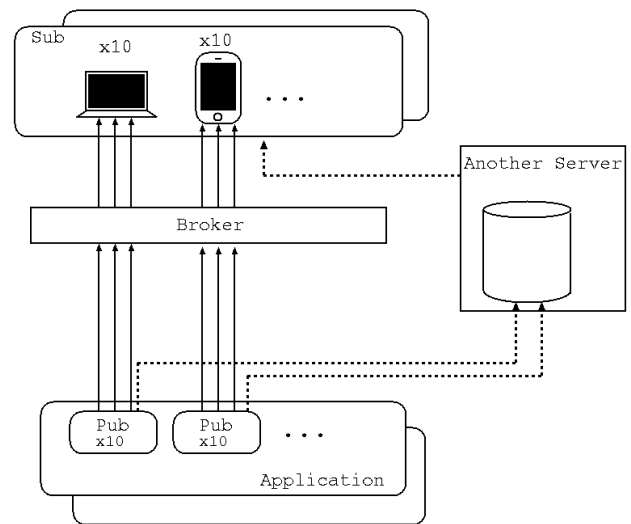


図 6 検証環境の構成

3.4 実装計画

前章で議論したシステム構成の実装の流れを述べる。まず、ブローカの処理するポインタとタイムスタンプの紐付けおよびデータ本体を格納するストレージを実装する。この際、ブローカとは別で用意したサーバの占める保存領域の管理やタイムスタンプの時刻の正確性を考慮する。

システム構成を図 6 に示す。実装したシステムを検証環境において動作させ、いくつかの適用例において有効であることを確認する。

動作の検証として、Pub とブローカのサーバ間にクロックのズレがある可能性を考え、等時性を保持したデータや途中で不必要になったデータなどを処理する際の変化に対する対応を考察する。また、既存の Pub/Sub システムと具体的な性能について比較し、スケーラビリティについても優位性を検証する。

現時点で構成したシステムの実装を行った後、そこから得られた結果をもとに高度化に関わる要素の再検討を行う。

4. おわりに

本稿では、Pub/Sub 機構において Pub/Sub 機構における通信形態や配送するデータの意味を保持しつつも、同期性や等時性という特徴を有したデータの配送を可能にし、データの種類ごとに適した通信の確立するシステムを提案した。まず、Pub/Sub におけるデータ配送および、データの特徴を整理し、それらを考慮した高度化に関する項目を検討した上で手法を示した。1 つ目は、ポインタである。ブローカにはポインタのみを流し、データ本体は別の場所から得るという構成になる。これにより、ブローカへの処理負担が軽減される効果が見込まれる。2 つ目に、タイムスタンプを用いた構成を示した。タイムスタンプを用いることで古くなったデータを削除することができ、不必要なデータを容易に扱うことができる。

さらに、タイムスタンプとポインタを組み合わせた提案

システムを新たに示した。このシステムでは、ブローカはポインタとタイムスタンプのみを扱う。この仕組みであれば、いくつかのデータ管理ストレージを組み合わせることができ、スケーラビリティも保証されると期待できる。また、ポインタにより、他ブローカへのリダイレクトも可能となるため、データ本体も扱う様なブローカや、異種ブローカとの連携も可能にし、今回のシステムをデータ共有ハブとして位置付けることもできる。加えて、実世界で収集した観測データなどの情報をコンピュータ技術を活用し、サイバー空間で解析することで、解析から得られる情報を産業の活性化や社会問題の解決を図るサービスおよびシステムのような柔軟かつ拡張性に優れた共通基盤において、多様なサービスをつなげるための仕組みとしても有効であると期待できる。

今後は概念実証としてサービスを提供するアプリケーションに構成したシステムを実装し動作の検証を行う。また、そこから得られるフィードバックをもとに、構成したシステムにおける要素と既存手法との比較を行う。

参考文献

- [1] V.Jacobson, D.K.Smetters, J.D.Thornton, M.F.Plass, N.H.Briggs, and R.L.Braynard: Networking Named Content, International Conference on Emerging Networking Experiments and Technologies, pp. 1-12, (2009).
- [2] P.T. Eugster, P.A. Felber, R. Guerraoui, and A.M. Kermarrec: The Many Faces of Publish/Subscribe,” ACM Computing Surveys, vol.35, no.2, pp.114-131, (2003).
- [3] MQTT, 入手先 (<http://mqtt.org>) (参照 2020-02-21).
- [4] Amazon Web Services: Designing MQTT Topics for AWS IoT Core, May 2019.
- [5] 坂野遼平, 竹内亨, 武本充治, 神林隆, 川野哲生, 松尾真人: ストリームデータの配送に向けた分散トピックベース Pub/Sub 手法の提案, 信学技報, vol. 113, no. 364, IA2013-64, pp. 41-46 (2013).
- [6] M.Castro, P.Druschel, A.M.Kermarrec, and A.Rowstron: SCRIBE: A large-scale and decentralized application-level multicast infrastructure, IEEE Journal on Selected Areas in communications, vol. 20, no. 8, pp. 1489-1499, 2002.
- [7] 内山仁, 峰野博史: 優先度を考慮した送信制御が可能な P-MQTT の開発と評価, マルチメディア, 分散協調とモバイルシンポジウム 2019 論文集, pp. 64-70 (2019).
- [8] R.Banno, S.Takeuchi, M.Takemoto, T.Kawano, T.Kambayashi, and M.Matsuo: Designing Overlay Networks for Handling Exhaust Data in a Distributed Topic-based Pub/Sub Architecture, Journal of Information Processing, vol. 23, no. 2, pp. 105-116 (2015).
- [9] S.Q.Zhuang, B.Y.Zhao, A.D.Joseph, R.H.Katz, and J.D.Kubiatowicz: Bayeux : An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination, Proc. International Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 11-20 (2001).
- [10] Nen-Fu.Huang, Rong-Tai.Liu, Chih-Ho.Chen, Ying-Tsun.Chen, Li-Wen.Huang: A Fast URL Lookup Engine for Content-Aware Multi-gigabit Switches, 2009 International Joint Conference on Computational Sciences and

Optimization, Sanya, Hainan, 2009, pp. 343-346 (2009).