一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IA2013-64(2013-12)

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

坂野　遼平† 竹内　亨† 武本　充治† 神林　隆† 川野　哲生†
松尾　真人†

† 日本電信電話株式会社 NTT 未来ねっと研究所
東京都武蔵野市緑町 3-9-11
E-mail: †{banno.ryohei, takeuchi.susumu, takemoto.michiharu, kambayashi.takashi, kawano.tetsuo, matsuo.masato}@lab.ntt.co.jp

あらまし　本研究では，トピックベースの pub/sub メッセージングをオーバレイネットワークを用いて自律分散的に構成する手法を検討する．publish される対象として映像やセンサ値といった実空間情報のストリームを想定した場合，ネットワーク資源の消費が激しいことが予想される．従って，オーバレイネットワーク上での配送経路長を可能な限り抑え，また subscriber が不在のトピックでは publish を停止できることが望ましいが，従来の手法では実現が困難であった．そこで本研究では，オーバレイネットワークに関する強 relay-free の概念を新たに定義し，Skip Graph を用いて強 relay-free 性を満たすトポロジを構築することで，ストリームデータの配送に適した分散 pub/sub 手法を提案する．シミュレーションプログラムを用いた実験により，配送経路長，負荷分散の公平性，publish の停止性の観点から提案手法が既存手法に対し優位であることが明らかとなった．
キーワード　トピックベース pub/sub，構造化オーバレイ，Skip Graph

A Distributed Topic-based Pub/Sub System for Stream Data Delivery

Ryohei BANNO†, Susumu TAKEUCHI†, Michiharu TAKEMOTO†, Takashi KAMBAYASHI†,
Tetsuo KAWANO†, and Masato MATSUO†

† NTT Network Innovation Laboratories, NTT Corporation
3-9-11, Midori-cho Musashino-shi, Tokyo
E-mail: †{banno.ryohei, takeuchi.susumu, takemoto.michiharu, kambayashi.takashi, kawano.tetsuo, matsuo.masato}@lab.ntt.co.jp

Abstract　We propose a new method of distributed topic-based pub/sub messaging using structured overlay networks. Supposing stream data of real-space information like sensor data and videos as objects to be published, the consumption of network resources is considered intense. It is preferable that the publication can be stopped in topics which have no subscribers and the path length of publication is short as possible, however conventional methods have difficulty of fulfilling these requirements. In this study, we define a new concept about overlay networks named “strong relay-free”, and propose a new method of distributed pub/sub suitable for stream data using the Skip Graph to build a topology which satisfies the strong relay-free property.
Key words　Topic-based Pub/Sub, Structured Overlay Network, Skip Graph

1. はじめに

Internet of Things（IoT）による高度に情報化された社会の実現に向けて，多様なサービスを構築可能な基盤が求められている．そこで，様々な異種要素を連携可能とする柔軟性や拡張性を備えたサービス基盤として，我々は Agent-based Service Platform（ASPF）の研究を進めている [1].

ASPF では，各種センサやアクチュエータをエージェントとして表現し，それらエージェントが広域に分散してネットワークワイドに連携することを想定している．そのような連携を

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，目的やデータ内容に応じたコンテンツ指向の通信 [2] 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ングパラダイムとして，publish/subscribe 型の通信（pub/sub）が注目されている．Eugster らによれば，pub/sub とは以下 3 点の性質を満たす通信である [3]．

- Space decoupling : 送受信者が互いの知識を持たない．
- Time decoupling : 送受信者が同時にアクティブである必要がない．
- Synchronization Decoupling : 送受信は非同期的に実行される．

これらの性質は，ASPF においてイベント駆動型のサービスを実現する上でも不可欠なものである．

一般的な pub/sub では，アプリケーションが出すコンテンツ指向の要求と，実ネットワークにおける位置指向通信との対応付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がある．IoT におけるデバイス数は 2020 年に 1,000 億台に達する [4] とも言われており，送受信者共に膨大な数となることが想定されるため，サービス基盤としては高いスケーラビリティが求められる．特に近年では，カメラ映像等の大容量データを手軽に生成できる環境の普及とともに，それらデータを活用した TwitCasting やニコニコ生放送のようなサービスが広がりを見せている．IoT においてこのような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ことが考えられる．

こうした観点から，pub/sub を非集中的に実現する分散 pub/sub が注目を集めている．本稿では，pub/sub の一類型であるトピックベース pub/sub に着目し，構造化オーバレイネットワークの技術を用いることで，ストリームデータの配送に適した分散 pub/sub の手法を提案する．既存手法として分散ハッシュテーブル（Distributed Hash Table, DHT）[5] を応用した方法 [6] 等が存在するが，配送経路の長さ，負荷分散の公平性，及び subscriber 不在時のネットワーク資源利用効率の観点で問題がある．そこで提案手法では，トピックベース pub/sub における relay-free の概念 [7] を拡張し，構造化オーバレイのアルゴリズムである Skip Graph [8] を応用することで，上記問題点を解消する手法を提案する．

2. トピックベース pub/sub と分散化手法

トピックベース pub/sub は，トピックと呼ばれる論理的なチャネルを介して publisher から subscriber へのメッセージ配送がなされるスキームである [3]．

本章では，トピックベース pub/sub に関する既存の分散化手法として Scribe [6] 及び PolderCast [9] について述べる．

2.1 Scribe

Scribe は，DHT を用いた Application Layer Multicast（ALM）の手法である．DHT では，あるノードがキーを指定して探索を行なうと，そのキーのハッシュ値を担当するノードが一意に定まり，探索クエリが届く（key-based routing [10]）．従って，DHT において複数のノードから同一のキーで探索を実行すると，どのノードのクエリも同一のノードへと辿り着く．そのノードを根ノードとして，各ノードからの探索経路を逆向きに辿ると，木構造と見做すことができる．Scribe はこの木構造をマルチキャストツリーとして用いる．

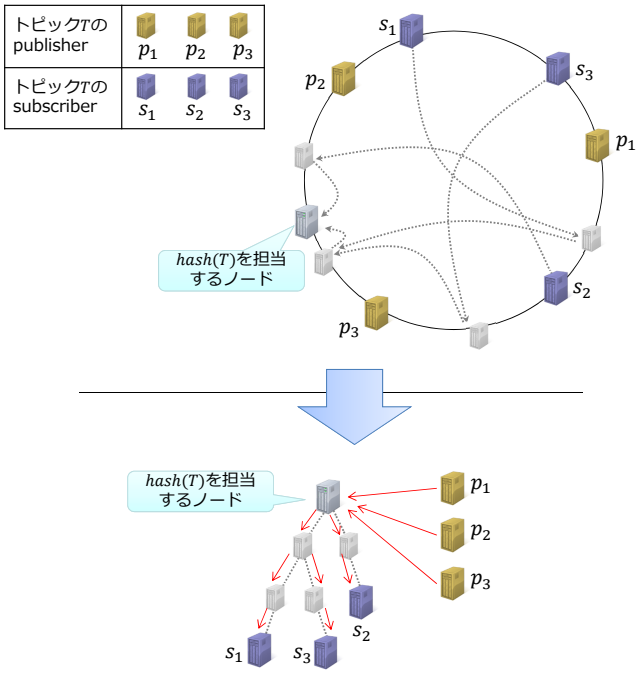

図 1　Scribe によるトピックベース pub/sub

トピックベース pub/sub に応用する場合，トピック毎に subscriber を葉ノードとする木構造を構築し，また publisher はトピック名をキーとして探索することで根ノードの宛先情報（IP アドレス等）を得てキャッシュしておく．publish 時は当該根ノードへと送信し，以降木構造に沿って転送することで全 subscriber にメッセージが届く（図 1）．

2.2 PolderCast

PolderCast は，ゴシップベースの P2P 型トピックベース pub/sub 手法である．Rings, Vicinity, Cyclon の 3 つのモジュールから成る階層構造となっており，Rings モジュールはトピック毎にリング状のオーバレイネットワークを形成する．Vicinity モジュールはノード間の類似性に基づくランダムリンクを構成し，Cyclon モジュールは一様なランダムリンクによって全ノードが接続されたオーバレイネットワークを維持する．即ち，PolderCast は非構造化オーバレイネットワークをベースとしてトピック毎のリングとショートカットリンクを構築する手法であると言える．

PolderCast では，subscriber はトピックを指定して当該トピックのリングに参加する．publisher も同様にトピックを subscribe することでリングに参加し，publish 時はリング内を対象としたメッセージのフラッディングを行なう．フラッディングに際しては，リング内を線形に転送することに加え，Vicinity 及び Cyclon モジュールによるショートカットリンクを用いることで，配送経路長を短縮することができる．

2.3 既存手法の問題点

これら既存手法には，以下に挙げる問題点が存在する．

a） 配送経路長

Scribe の場合，publisher から各 subscriber までの配送経路長は DHT の探索経路長と等しいオーダとなるため，全ノード数を N とすれば $O(\log N)$ である．常に全体のノード数に依存した経路長を要するため，例えば subscriber 数が 1 のトピックであっても何回もの転送を経てメッセージを届けることとなり，ネットワーク資源の消費及び到達遅延の観点で望ましくない．

b） 負荷分散の公平性

Scribe では，各ノードが自身とは無関係なトピックにおけるメッセージの中継を担う．ASPF のような基盤上では様々なサービスが共存するため，トピック毎の publish 頻度に差がある状況を想定する必要があるが，この場合，例えば「1 日 1 回程度 publish される温度情報を得る目的で参加しているノードが，映像ストリームの転送を強いられる」といった負荷分散の不公平性が生じ得る．このため，個々のノードが自身の負荷を事前に見積ることが難しくなる問題がある．

c） publish の停止性

PolderCast 及び Scribe では，subscriber の不在時にも publish を停止することができず，ネットワーク資源を無駄に消費してしまう問題がある．PolderCast の場合，publisher もトピックを subscribe することでリングに参加するため，subscriber が不在の場合であっても publisher が存在する限りフラッディングが生じる．Scribe では，publisher は根ノードに対し常に publish し続ける必要がある．

3. 分散トピックベース pub/sub における relay-free 性

3.1 relay-free 性の定義と特徴

2.3 節に述べた問題点を改善するために必要な性質として，relay-free 性 [7] が挙げられる．relay-free 性とは，各トピックのメッセージ配送処理に対し，当該トピックに関心を持つノードのみが関与する性質である．即ち，オーバレイネットワークをグラフとして捉えた時，各トピックについて publisher 及び subscriber であるノードのみで構成される部分グラフが連結である状態を指す性質であり，Topic-connected Overlay（TCO）とも呼ばれる．relay-free 性の定義を以下に述べる．

> ノード集合 V，トピック集合 T が与えられた時，ノード $v \in V$ 及びトピック $t \in T$ を入力とするブール値関数 $Int(v,t)$ を考える．$Int(v,t) = true$ の時，v は t に関心を持つノード，即ち t の subscriber または publisher であるとする．オーバレイネットワークをグラフ $G = (W, E)$ と表す．ここで，$W = V$，$E \subseteqq V \times V$ である．$\forall t \in T$ について，ノード集合 $\{v \in V | Int(v,t) = true\}$ が誘導する G の部分グラフが連結であるとき，G は relay-free である．

relay-free なオーバレイネットワークでは，一般に，配送経路長がトピックのサイズ（当該トピックの publisher 及び subscriber の数の合計）に依存し，小さなトピックでは配送経路長が短くなる．また，無関係なノードがメッセージ転送に関与しないという特徴がある．2. 章に述べた PolderCast は relay-free 性を備えているため，Scribe と比べ配送経路長が短く，負荷分散の不公平性も生じない利点がある．

3.2 強 relay-free 概念の提案

relay-free 性によって前述のような利点が得られるが，2.3 節に挙げた問題点のうち publish の停止性については容易には実現できない．各 publisher が存在を把握し得るのは自身が直接リンクを持つノードのみであり，自身とはリンクを持たない subscriber の存在/不在を判断することが難しいためである．

そこで本稿では，publish の停止性についても考慮し relay-free 性を拡張した**強 relay-free 性**の概念を提案する．従来の relay-free 概念においては，publisher と subscriber はいずれもトピックへの参加ノードとして同等に扱われる前提となっているが，強 relay-free ではこれらの区別を導入する．なお，従来の relay-free 性を本稿では**弱 relay-free 性**と呼ぶこととする．強 relay-free 性の具体的な定義を，以下に述べる．但し，V，T，G の定義は前述の弱 relay-free の場合と同様である．

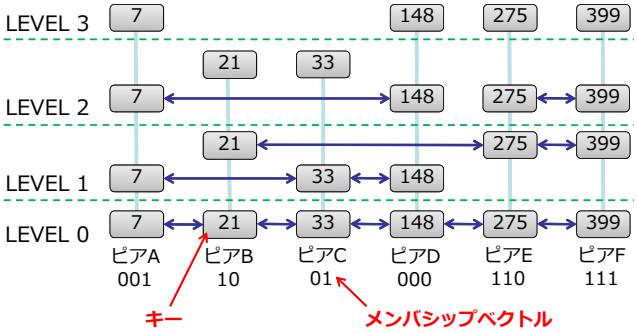

図 2 Skip Graph の構成例

> ノード $v \in V$ 及びトピック $t \in T$ を入力とするブール値関数 $Sub(v,t)$ 及び $Pub(v,t)$ を考える．$Sub(v,t) = true$ の時，v は t の subscriber であり，$Pub(v,t) = true$ の時，v は t の publisher であるとする．$\forall t \in T$ について，ノード集合 $\{v \in V | Sub(v,t) = true\}$ が誘導する G の部分グラフが連結であり，かつノード集合 $\{u \in V | Sub(u,t) = true \cup Pub(u,t) = true\}$ が誘導する G の部分グラフが連結であるとき，G は強 relay-free である．

強 relay-free なオーバレイネットワークにおいては，各トピックの suscriber は連結部分グラフを構成しているため，subscriber の存在/不在をひとつの部分グラフの存在/不在と捉えることができる．従って部分グラフを仮想的なひとつのノードのように捉えることで，部分グラフとのリンクを持つノードが当該トピックに関する subscriber 全体の存在/不在を判断できる可能性が得られる．本稿では，この強 relay-free 性を用いた効率的な分散トピックベース pub/sub 手法を提案する．

4. publish 停止可能なトピックベース pub/sub 手法

本章では，提案手法について述べる．まず，構造化オーバレイのアルゴリズムである Skip Graph [8] 及びその派生である Multi-key Skip Graph [11] について述べた後に，これらのアルゴリズムを応用した提案手法の内容について説明する．

4.1 要素技術

4.1.1 Skip Graph

Skip Graph では，各ノードが 1 つずつキーと呼ばれる情報を持ち，任意のノードからキーの値または範囲を指定した検索を行なうことができる．Skip List [12] を多重化した構造を持っており，図 2 に示すような階層を形成する．

最下位層は，キーをソートした順序でノードが並んだ双方向リストとなっている．各ノードはキーとは別にメンバシップベクトルと呼ばれる k 進数の乱数列（本稿では $k = 2$ とする）を持ち，レベル i ではメンバシップベクトルの接頭 i 桁までが等しいノード同士で双方向リストを形成する．

検索を行なう時は，検索開始ノードの最上位レベルからス

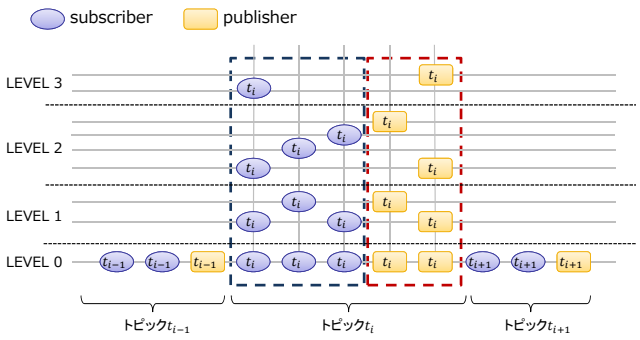

図 3　各トピック内における publisher と subscriber の順序付け

タートし，Skip List と同様に目的のキーを通り過ぎない範囲でホップしてレベルをひとつ下げるという動作を繰り返す．これにより，上位レベルがショートカットリンクとして働き，目的キーまで少ないホップ数で到達することができる．

Skip Graph において各ノードが保有する経路表のサイズと，検索時の経路長は，ノード数を N とした時，共に $O(\log N)$ である．

4.1.2　Multi-key Skip Graph

Multi-key Skip Graph は，Skip Graph において各ノードが複数のキーを保持可能とした拡張アルゴリズムであり，各ノードのキーは仮想ピアとして挿入される．ノード毎の仮想ピア数を m，ノード数を N とした時，Multi-key Skip Graph における経路表のサイズは $O(m \log N)$，検索時の経路長は $O(\log N)$ となる．

4.2　提 案 手 法

4.2.1　強 relay-free 性を満たすトポロジの構築

各トピックの publisher 及び subscriber にトピック名をキーとして保持させ，Multi-key Skip Graph を構成することで，トピックベース pub/sub を実現することができる [13]．この時，同一トピックの publisher 及び subscriber は Multi-key Skip Graph のレベル 0 において隣接することから，弱 relay-free 性を備えていると言える．

提案手法では，トピック名をキーとしてノード間の順序を定めることに加え，publisher と subscriber との間にキー（トピック名）よりも低いプライオリティで全順序関係を持たせる．例えば，トピック名に対し，publisher と subscriber とで異なる予約語を接尾辞として付した上でキーとして用いることで，そのような順序関係を実現することができる．これにより，Multi-key Skip Graph のレベル 0 リストでは，各ノードがトピック単位でソートされた上で，さらに各トピック内でノード種別（publisher または subscriber）によってソートされている形となる（図 3）．

同一トピックの publisher 及び subscriber は隣接しており，また subscriber 同士も全て隣接しているため，このトポロジは強 relay-free 性を備えている．トピック $t \in T$ において，subscriber が隣接する領域を $SEG_{sub}(t)$，publisher が隣接する領域を $SEG_{pub}(t)$ と表すこととする．

ここで，$\exists t \in T$ について，publisher が 1 つ以上存在する場合は，レベル 0 において $SEG_{sub}(t)$ と隣接する publisher ノードがただ 1 つ存在する．このノードをランデブーポイントと呼び，$rp(t)$ と表す．$rp(t)$ は，トピック t に関するメタ情報を保持することなく，トピックの subscriber が存在するか否かを判断することが可能である．即ち，レベル 0 において $SEG_{sub}(t)$ 側の隣接ノード v について $Sub(v,t) = true$ であれば subscriber は存在し，$Int(v,t) = false$ であれば存在しない．

$rp(t)$ と $SEG_{sub}(t)$ の間に新たに publisher が挿入された場合は，新たな publisher が $rp(t)$ として振る舞う．また $rp(t)$ が離脱した際は，レベル 0 において $rp(t)$ と $SEG_{pub}(t)$ 側で隣接していた publisher ノードがいれば，そのノードが新たな $rp(t)$ となる．

4.2.2　publish の停止及び再開

subscriber の存在/不在をランデブーポイントが他の publisher へ通知することで，subscriber 不在時の publish の停止を実現する．トピック t について，subscriber が全て離脱し 0 となった場合，$rp(t)$ は経路表の更新を監視しておくことで，それを受動的に検知することができる．subscriber の不在を検知した $rp(t)$ は，$SEG_{pub}(t)$ に対し publish 停止の指示をマルチキャストする．また，subscriber が不在のトピックに新たに subscriber が入ってきた場合も，同様にして受動的な検知が可能であり，$rp(t)$ が $SEG_{pub}(t)$ に対し publish 開始の指示をマルチキャストする．

4.2.3　publish，subscribe，unsubscribe

トピック t について publish を実行する際は，Multi-key Skip Graph を用いた $SEG_{sub}(t)$ に対するマルチキャストを行なう．また subscribe 及び unsubscribe 処理は，Multi-key Skip Graph における仮想ピアの挿入及び削除の処理に準ずる．

4.2.4　publisher の挿入

あるトピックについて新たに publisher が参加する場合，参加直後から publish を行なうべきか否かを判断する必要がある(注1)．参加先トピックに既に publisher が存在する場合，Multi-key Skip Graph へのノード挿入処理の過程で少なくとも 1 つの既存 publisher とメッセージのやり取りが行われるため，そのメッセージに publish 停止中か否かの情報を上乗せし，新規 publisher が publish 開始の判断を行なう．publisher が存在しない場合は，自身がランデブーポイントとなるため，挿入完了後に publish 開始を自己判断する．

4.2.5　提案手法の性質

トピックへの参加ノード数を m とした時，提案手法における配送経路長は $O(\log m)$ となる．また提案手法では，無関係なトピックの publish 転送を担わされることが無く，負荷分散の不公平性が生じない．さらに，subscriber が不在のトピックでは publish を停止することが可能である．

4.3　不整合の解消

提案手法では，急なノードの離脱によって，以下 2 パターンの不整合が生じ得る．

（1）　subscriber が不在時に publish が行われる状況

（2）　subscriber が存在時に publish 停止状態になる状況

いずれもランデブーポイントでは発生することはなく，その他の publisher ノードにおいて発生の可能性がある．以下，これら不整合の解消策について述べる．

(1) の不整合について，トピック t において $rp(t)$ 以外の publisher が subscriber 不在時に publish を行った場合，そのメッセージは必ず $rp(t)$ を経由する．従って，$rp(t)$ が他の publisher

(注1)：あるトピックについて publisher であると同時に subscriber でもあるノードについては，subscriber としてのキーのみを Multi-key Skip Graph に挿入する．こういったノードは，自身が存在する限り publisher も subscriber も存在する状況となり，subscriber の不在検知や他の publisher への通知といった処理が不要なためである．

からメッセージを受け取った場合，subscriber の不在を確認した上で送り元の publisher に publish 停止の指示を送信する．

一方 (2) の不整合については，publish 停止中の各 publisher が，Multi-key Skip Graph のレベル 0 リストにおける隣接ノードに対し定期的に publish 停止状態か否かを問い合わせる．問い合わせ結果と自身の状態とに齟齬があった場合，$rp(t)$ に publish 開始の指示を出すよう要求するメッセージを送り，$rp(t)$ は subscriber の存在を確かめた上で，$SEG_{pub}(t)$ に対し publish 開始の指示をマルチキャストする．

5. 評　　価

2.3 節において，既存手法の問題点として配送経路長，負荷分散の公平性，publish の停止性の 3 点を挙げた．本章では，これら 3 点について行った評価実験について述べる．

実験に際し，比較対象として 2. 章に述べた Scribe を用いた(注2)．Skip Graph ではメンバシップベクトルを対象としたルーティングが可能 [14] であり，これを用いて DHT を構築することができる [15]．従って，Skip Graph ベースの DHT 上に Scribe を構築することで，提案手法と Scribe における経路表サイズ等の条件をある程度揃えて比較することができる．以降に述べる実験では，Java 言語で実装したシミュレーションプログラムを用いている．

5.1 配送経路長

publisher から subscriber までの平均経路長（publish 時のホップ数）について確認する実験を行った．以下，pub_t はトピック毎の publisher 数，sub_t はトピック毎の subscriber 数を表すものとする．具体的な実験設定として，トピックのサイズは 1,000 ノードと 10 ノードの 2 パターンを用意し，全トピックで共通とした．またトピック内における subscriber と publisher の構成として，$pub_t < sub_t$，$pub_t = sub_t$，$pub_t > sub_t$ の 3 パターンを想定し，上記トピックサイズとの組み合わせとして以下の 6 パターンを設定して，全体のノード数を 1,000，10,000，100,000 に変化させて経路長への影響をグラフにプロットした．

- $pub_t = 10$，$sub_t = 990$
- $pub_t = 1$，$sub_t = 9$
- $pub_t = 500$，$sub_t = 500$
- $pub_t = 5$，$sub_t = 5$
- $pub_t = 990$，$sub_t = 10$
- $pub_t = 9$，$sub_t = 1$

実験の結果を，図 4 に示す．プロットされている値は，各トピックにおいて各 publisher から subscriber までの平均経路長を算出し，全 publisher についてその平均をとったものである．

Scribe では全体のノード数が増加すると経路長も長くなっているが，提案手法では経路長がノード数には影響を受けていないことがわかる．また提案手法の場合トピックのサイズが小さいほど経路長が短くなっているが，Scribe ではトピックサイズにかかわらず概ね一定であり，全体として提案手法よりも経路長が長いことが見てとれる．例えば 100,000 ノード時，トピックサイズが 1,000 である 3 パターンに着目すると，提案手法がいずれも経路長 4 未満であるのに対し，Scribe では経路長が 16 を超えており，4 倍以上のホップ数を要することがわかる．

(注2)：PolderCast は非構造化オーバレイをベースとした手法であり，構造化オーバレイベースの手法とは想定する環境に差があること，また経路表サイズ等の条件を揃えた比較が難しいことから，本稿では関連研究としての言及に留めている．

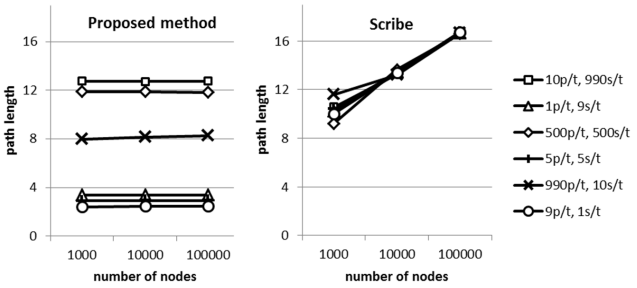

図 4　配送経路長の比較

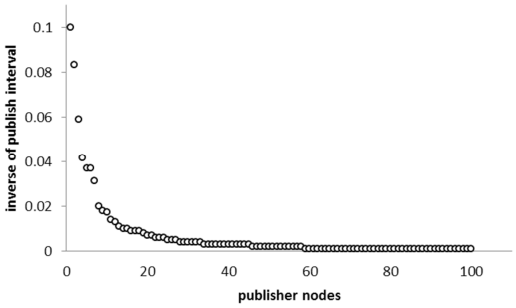

図 5　publish 頻度の分布

5.2 負荷分散の公平性

publish に関するメッセージ転送の負荷がノード間でどのように分散しているかを確認する実験を行った．

実社会における pub/sub の一例として，Twitter が挙げられる．Twitter は各ユーザをトピックと捉えればトピックベースの pub/sub とみなせるが，ユーザ毎の発言頻度はべき乗則に従うという報告がある [16]．そこで本実験では，各トピックの publish 頻度を図 5 に示すようなべき乗則で算出した．少数のノードは publish の間隔が短く，最小で 10 msec であり，多くのノードは publish の間隔が長く，最大で 1,000 msec である．

このようにして publish 頻度を割り当てた publisher を 100 ノード立ちあげ，それぞれ異なるトピックに所属させた．そして各トピックに subscriber を 1,000 ノード用意した．即ち，トピック数は 100 でありノード数は 100,100 である．

この実験設定のもと，前述の頻度で各 publisher に publish を実行させ，全体の publish 実行回数が 1,000 回に達するまで，各ノードにおけるメッセージの送信回数，受信回数，転送回数をカウントした．なお，送信回数とは publisher における publish の回数，受信回数とは subscriber が subscribe しているトピックの publish を受信した回数，転送回数とは送信を除く他ノードへのメッセージ転送の回数である．全ノードを転送回数の多い順にソートした上で 1,000 ノード毎にグループ化し，各回数の平均値を算出してプロットしたものが図 6 である．

Scribe では，送受信回数にかかわらず転送回数が非常に多くなっているケースが存在する．一方，提案手法の場合受信回数と転送回数に相関が現れており，相関係数を算出したところ提案手法では 0.5885，Scribe では −0.0045 であった．これは，提案手法では subscribe するトピックの publish 頻度に応じてメッセージの転送回数が増減することを意味しており，Scribe と比べ負荷分散の不公平性が生じていないと言える．また，グラフからは，全体として提案手法の方が転送回数が少なく抑えられることも見てとれる．なお，提案手法のグラフ内右側にお

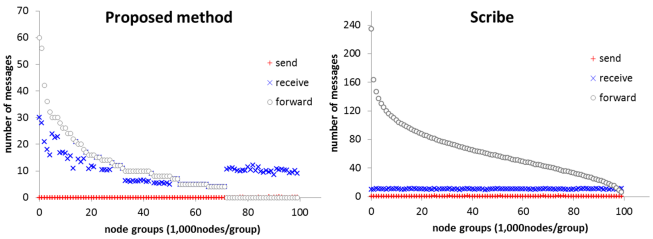

図 6　各ノードのメッセージ送受信・転送回数（1,000 ノード毎の平均）

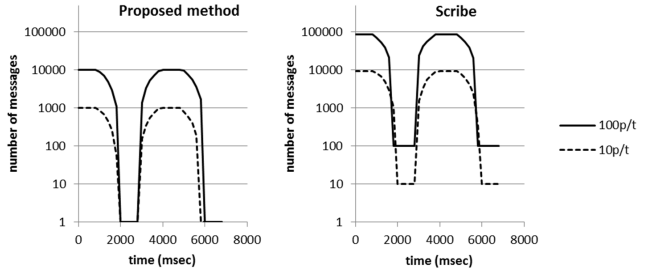

図 7　publish によるメッセージ数

いて，転送回数が 0 であり受信回数が 10 程度となっている範囲がある．これは各トピックの配送木上で葉ノードとなったノード群であり，提案手法では強 relay-free 性によって無関係なトピックの転送を行なうことが無いためこのようなノード群が現れる．

5.3　publish の停止性

トピックに subscriber が存在しない場合の publish の停止性について確かめる実験を行った．

全体のノード数を 100,000 とし，トピック毎の publisher 数について $pub_t = 100$ 及び $pub_t = 10$ の 2 パターンを設定した．各 publisher は 200 msec 間隔で publish を続けるものとし，あるトピックにおいて subscriber が以下のシナリオで増減を繰り返す設定とした．

（1）　100 ノードでスタートし 1,000 msec 間そのまま．
（2）　10 msec 間隔で 1 ノードずつ unsubscribe．
（3）　0 ノードになったら 1,000 msec 間そのまま．
（4）　10 msec 間隔で 1 ノードずつ subscribe．
（5）　100 ノードになったら 1,000 msec 間そのまま．
（6）　10 msec 間隔で 1 ノードずつ unsubscribe．
（7）　0 ノードになったら 1,000 msec 間そのまま．

当該トピックの publish に関してオーバレイネットワーク上で転送されるメッセージ数をカウントした結果を，図 7 に示す．なお，グラフの縦軸を対数としたため，数値が 0 の部分には 1 を加えて表示している．

提案手法では subscriber が 0 になった期間はメッセージ数が 0 になっているが，Scribe では subscriber が不在の期間もオーバレイネットワーク上をメッセージが流れていることがわかる．

6.　おわりに

本稿では，ストリームデータを想定した場合に望ましい性質としてオーバレイネットワークにおける強 relay-free の概念を新たに定義し，構造化オーバレイのアルゴリズムである Skip Graph を用いることで，強 relay-free 性を満たす分散トピックベース pub/sub の手法を提案した．提案手法は relay-free 性を持たない手法と比べ publish に関する配送経路長が短く，また各ノードが無関係なトピックの publish 転送を担わされないため，負荷分散の不公平性が生じないという利点がある．さらに，subscriber が不在のトピックでは publish を停止することが可能であり，ネットワーク資源の消費を抑えることができる．

今後の課題として，不整合の解消に関する検証や実環境での検証を含む実験を実施する予定である．また，Twitter 等の実データを用いた評価も行いたいと考えている．

文　献

[1] 竹内　亨，坂野遼平，馬越健治，川野哲生，神林　隆，武本充治，松尾真人，柿沼隆馬，“BMI 応用サービスの実現に向けたエージェントベース分散処理基盤の提案と評価，” 情報処理学会研究報告，vol.2013-DPS-1，no.9，pp.1–8，2013.

[2] 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.

[3] 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.

[4] S. Hodges, S. Taylor, N. Villar, and J. Scott, “Prototyping Connected Devices for the Internet of Things,” IEEE Computer, pp.26–34, 2013.

[5] A.I.T. Rowstron and P. Druschel, “Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems,” IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing, no.November 2001, pp.329–350, 2001.

[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] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, “Constructing Scalable Overlays for Pub-Sub with Many Topics,” ACM Symposium on Principles of Distributed Computing, pp.109–118, ACM Press, 2007.

[8] J. Aspnes and G. Shah, “Skip Graphs,” ACM Transactions on Algorithms (TALG), vol.3, no.4, pp.37:1–37:25, 2007.

[9] V. Setty, M.V. Steen, R. Vitenberg, and S. Voulgaris, “PolderCast: Fast, Robust, and Scalable Architecture for P2P Topic-based Pub/Sub,” International Middleware Conference, pp.271–291, 2012.

[10] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica, “Towards a Common API for Structured Peer-to-Peer Overlays,” International workshop on Peer-To-Peer Systems, 2003.

[11] 小西佑治，吉田　幹，竹内　亨，寺西裕一，春本　要，下條真司，“単一ノードに複数キーを保持可能とする Skip Graph 拡張，” 情報処理学会論文誌，vol.49，no.9，pp.3223–3233，2008.

[12] W. Pugh, “Skip Lists : A Probabilistic Alternative to Balanced Trees,” Communications of the ACM, vol.33, no.6, pp.668–676, 1990.

[13] 坂野遼平，竹内　亨，武本充治，松尾真人，“ユビキタスサービスプラットフォームにおける Skip Graph を用いた publish/subscribe 機構の検討，” 電子情報通信学会総合大会講演論文集，通信 (2)，no.B-6-65，p.65，2013.

[14] N.J.A. Harvey, J. Dunagan, M.B. Jones, S. Saroiu, M. Theimer, and A. Wolman, “SkipNet: A Scalable Overlay Network with Practical Locality Properties,” USENIX Symposium on Internet Technologies and Systems, p.9, 2003.

[15] PIAX Inc., “PIAX: P2P Interactive Agent eXtensions”. `www.piax.org`（参照 2013-11-25）.

[16] A. Welhuis, “Twitter and the pareto principle”. `www.annouckwelhuis.nl/twitter-and-the-pareto-principle-2`（参照 2013-11-25）.