4X-05

ブラウザ間構造化P2Pネットワークを用いたマイクロブログサービスの設計と実装

田中凌，安倍広多
大阪市立大学大学院創造都市研究科

1 はじめに

Twitterに代表されるマイクロブログサービスが情報発信の手段として普及している．しかし一般的にマイクロブログサービスはサーバクライアント方式で実現されているため，災害などによってサーバが停止するとサービス全体が停止する問題がある．一方，主要なWebブラウザにおいてWebRTC（2つのWebブラウザ間で直接通信するための規格）の実装が進み，Webブラウザによる P2P ネットワークが実現できるようになっている．このため，本研究ではWebブラウザ上で動作する Pure P2P 型のマイクロブログサービス **Kizuna** を開発する．Webブラウザ上で実現することで，専用アプリケーションのインストールが不要となり，ユーザの利便性は高い．

著者らは文献 [1] で（Webブラウザベースではない）P2P ネットワークによるマイクロブログサービスを提案しているが，本研究はこれをベースに改良を加えたものである．Kizuna ではメッセージ発信，フォロー，メッセージ保存・取得など，Twitterのサブセットの機能を実現する．本稿では Kizuna の設計と実装について述べる．

2 Kirin

Kizuna は著者らが開発しているブラウザ間構造化 P2P ネットワーク Kirin[2] 上で実現する．Kirin は構造化 P2P ネットワーク Suzaku[3] を Web ブラウザ向けに改良したものであり，JavaScript (実際は TypeScript) で実装されている．ブラウザ間の通信には WebRTC を用いる．Kirin は主要な Web ブラウザと Node.js 上で動作する（Node.js は WebRTC のシグナリング処理に用いる）．Kirin はリングベースの構造化 P2P ネットワークであり，リング上でノード（論理ノード）は保持するキーの昇順に並ぶ．キーは文字列型である．1つの物理ノードは複数の論理ノードを挿入できる．Kirin はキーの範囲を指定したアプリケーションレベルマルチキャスト (ALM) をサポートする．ALM のアルゴリズムは文献 [4] と同様であり，要する最大ホップ数は経路表が収束している場合は $\lceil \log_2 n \rceil - 1$ である（n は論理ノード数）．Kirin は分散ハッシュテーブル（DHT）の機能も備えている．

3 Kizunaの設計

Kizuna の主要な設計目標は以下の通りである：(1) サーバに依存しない，非集中型の構造とする（ただし，WebRTC で用いる STUN サーバへの依存は残る），(2) 第三者によるメッセージの偽造を防止する，(3) オフライン中のユーザのメッセージを取得できる．

3.1 アカウント

P2P システムではメッセージを容易に偽造できるため，Kizuna ではデジタル署名を用いて発信者を検証する．このため，アカウント作成時に公開鍵と秘密鍵のペアを生成する．

Kizuna は非集中型とするため，キーサーバなどは使用せず，公開鍵は DHT に登録する（キーはユーザ名）．あるユーザ x が別のユーザ y をフォローする場合，まず y のユーザ名をキーとして DHT を検索し，y の公開鍵 *y.pub* を入手する．このとき，入手した公開鍵が実際に y 自身の公開鍵かどうかを検証する必要があるが，認証局の使用は利用者の負荷が高いため，Kizuna では入手した公開鍵の検証はユーザ自身の責任とした（例えば y は予め外部の信頼できる SNS や Web サイトなどに *y.pub* の指紋を掲載しておき，x はそれを確認するなど）．

公開鍵暗号の実装としては，JavaScript によるオープンソースの実装 (openpgp.js[1]) が利用できる PGP を用いた．このため，ユーザは自身が電子メールで利用している PGP の鍵をそのまま利用できる．また，Kizuna のユーザ名はメールアドレスとなる．

Kizuna の内部では，ユーザは keyid（公開鍵の指紋の下位 64 ビット）で識別する．

3.2 メッセージの配送

メッセージの配送は，ユーザによってはフォロワー数が数千人を超える可能性を考慮すると効率よく行う必要がある．Kizuna ではメッセージは Kirin の ALM で配送する．

ユーザ x は，オンラインとなったとき，Kirin のリング上でキーが $K(x)$ であるノード（**発信ノード**）を挿入する（ただし $K(x)$ は x の keyid）．また，ユーザ x のフォロワー y は，キーが $K(x)$+“.”+$K(y)$ であるノード（**購読ノード**）を挿入する．これによってキーが $K(x)$+“.” から始まる範囲に x のフォロワーの購読ノードが集まる．x がメッセージを発信する際は，x の発信ノードから，この範囲に ALM でメッセージを配送する（図 1）．

（購読ノードを通じて）y が x からのメッセージを受信した場合，y は x のフォロー時に入手した x の公開鍵を用いて署名を検証する．

3.3 メッセージの保存

ユーザ（y）がオフラインの間にも y がフォローしているユーザ（x）はメッセージを発信している．この

Design and Implementation of a Microblogging Service over Inter-Browser Structured P2P Networks
Ryo Tanaka and Kota Abe
Graduate School for Creative Cities, Osaka City Univ.

[1] https://github.com/openpgpjs/openpgpjs

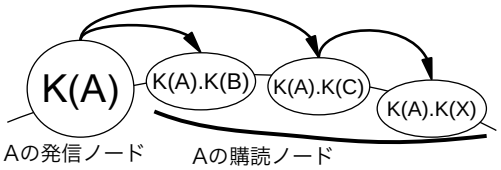

図 1: ALM によるメッセージ配送

メッセージを y がオンラインになった契機で取得できるように，すべてのメッセージはフォロワーへの配送と同時に DHT に格納することにした．しかし，すべてのメッセージにユニークなキーを付与してばらばらに DHT に格納すると，オンラインに戻る際，取得するメッセージ数だけ DHT get を実行する必要があり，時間を要する．

このため，Kizuna では DHT の値 (value) を配列化し，1 つのキーに複数の値を格納できるようにした．キーとしては各ユーザごとにユニークな値を使用し，配列のインデックスとしては，メッセージのシーケンス番号（あるユーザの各メッセージに 0 から順番に付与した値）を用いる．これにより，オンラインに戻る際，DHT の get をフォローするユーザ数だけ実行すれば良い．

3.4 DHT の改ざん防止

一般的な DHT では，put 操作を実行したノードから ⟨put 要求⟩ が指定されたキーを担当するノード（担当ノード）にルーティングされ，担当ノードがキーと値のペアを格納する．

Kirin では，DHT に書き込まれた値を権限がないユーザによって容易に変更（改ざん）されないようにするため，以下の対処を行った．

1. あるキーに対する最初の ⟨put 要求⟩ で，権限を持つユーザ (u) の公開鍵を担当ノードに登録する（Kirin の DHT では 1 つのキーに対して値を複数格納できるが，公開鍵は 1 つのキーに対して 1 つ登録する）．

2. 以降の同じキーに対する ⟨put 要求⟩ や ⟨remove 要求⟩ には u の署名を付与する．担当ノードは署名を登録されている公開鍵で検証し，正しい署名が付与されている要求だけを処理する．

なお，この方法は悪意あるノードが担当ノードになった場合には無力である．このような場合への対処は将来の課題である．

3.5 P2P ネットワークの構造

Kizuna が使用する Kirin の P2P ネットワークの構造を図 2 に示す．ここではユーザ A，B，X が参加し，A を B と X が，B を A が，X を A が，それぞれフォローしている．このため，対応する発信ノードと購読ノードが挿入されている．

各物理ノード（Web ブラウザ）は DHT を構成するために 1 つの DHT ノードを挿入する．DHT ノードのキーは “DHT.”+ノード ID（128 ビットの乱数）である．DHT に put する際は，キーのハッシュ値を超えない最大のノード ID を持つ DHT ノードに格納する．

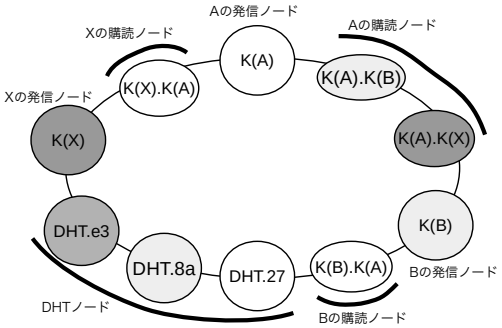

図 2: Kizuna における P2P ネットワークの構造

4 実装

Kizuna は現在実装中であるが，原稿執筆時点で DHT の値の配列化と改ざん防止，アカウント作成，メッセージ発信，フォロー，メッセージ保存機能などが動作している．実装言語としては TypeScript を用いている．PGP の実装としては（前述のとおり）openpgp.js を，また，Web アプリケーションフレームワークとしては Bootstrap を用いた．

Kizuna を利用するには次のステップが必要である．(1) 予め取得しておいた Kizuna のプログラム（HTML や JavaScript）をブラウザで開く（もしくはこれらを公開している Web サーバにブラウザでアクセスする），(2) ユーザのアカウント情報（ユーザ名，公開鍵，秘密鍵など），フォローしているユーザの情報（ユーザ名や公開鍵など），これまでに発信・受信したメッセージなどを格納した JSON ファイルを読み込ませる，(3) ブラウザ上のフォームで，Kizuna に参加済みの Node.js ノードの URL を入力し，これを introducer として Kizuna のネットワークに参加する．

5 おわりに

本稿では，P2P ネットワークを用いたマイクロブログサービス Kizuna の設計と実装について簡単に述べた．今後の課題としては全文検索機能の追加，定量的な評価などがある．

(謝辞) 本研究は JSPS 科研費 JP16K00135 の助成を受けている．

参考文献

[1] 播磨裕太, 安倍広多, 石橋勇人, 松浦敏雄 : KiZUNA: P2P ネットワークを用いた分散型マイクロブログサービスの実現, 情処研報, Vol. 2014-IOT-24, No. 18, pp. 1–6 (2014).

[2] 鄭焱祖, 川井悠司, 李　佼柯, 安倍広多 : WebRTC を用いた耐故障性の高いウェブブラウザ間構造化 P2P ネットワークの実現，情処研報，Vol. 2017-IOT-39/2017-SPT-25, No. 9, pp. 1–8 (2017).

[3] 安倍広多，寺西裕一 : 高い Churn 耐性と検索性能を持つキー順序保存型構造化オーバレイネットワーク Suzaku の提案と評価，信学技報，Vol. 116, No. 362 (IA2016-65), pp. 11–16 (2016).

[4] Banno, R., Fujino, T., Takeuchi, S. and Takemoto, M.: SFB: a scalable method for handling range queries on Skip Graphs, *IEICE Commun. Exp.*, Vol. 4, No. 1, pp. 14–19 (2015).