

メッセージフェリー方式を用いた 災害時通信アプリケーションフレームワークの提案と実装

長谷川 友香[†] 小口 正人[†]
[†]お茶の水女子大学

1. はじめに

近年、インターネットの普及に伴い、連絡手段として Facebook, Twitter 等の Web アプリケーションを用いるユーザも多い。しかし災害時にはネットワークインフラが断たれ、そのような方法で連絡が取れなくなる地域が生じる。本研究ではそのような環境下で Web アプリケーションを利用可能にする手法について検討する。

この課題を解決するための技術として遅延/途絶耐性ネットワーク (DTN: Delay/Disruption Tolerant Networking) が広く研究されている [1]。DTN とは、継続的なネットワーク接続が不可能な状況に耐えうる通信技術を広くとらえたものである。

本研究では広域的な災害発生時に利用可能な Web アプリケーション実現のための DTN 通信モデルと、それを用いた Web アプリケーションの実装方針を提案する。

2. 広域災害時に Web アプリケーションを利用するための手法の検討

本研究で言う Web アプリケーションとは、端末・サーバ間通信を行うアプリケーションのことである。そのため、インターネット接続不可能地域 (以下、オフライン地域) と接続可能地域 (以下、オンライン地域) でのメッセージのやり取りが必須となる。

2.1 災害時通信のための DTN ルーティング手法

DTN におけるルーティング手法は様々あるが、その中でも、Epidemic[2] などの確率的転送方式が DTN 研究の対象として広く注目を集めている。確率的転送方式とは、「うまく中継してくれそうな」端末にメッセージをホップする方式である。また、災害時の利用に焦点を絞った研究として、すれ違う端末間で経路表を交換することで確率的転送を行い、インターネットの接続できない地域からの Twitter の送受信を実現させる研究 [3] などがある。

確率的転送方式を用いた場合、まずリクエストをオフライン地域から DTN 端末をホップしてオンライン地域まで到達させる必要がある。仮にオフライン地域を半径 10km、端末間の無線通信距離を半径 50m とし、端末は均一に分布しているとしてオフライン地域の中心からオンライン地域までリクエストを転送することを考えると、単純計算で 200 ホップが必要となる。各端末間でのデータ転送成功確立を 95% とすると、リクエストがオンライン地域に到達する確率は約 0.0035% と非常に低くなる。また、リクエストがサーバに到達できたとしても、サーバはどの端末にレスポンスを転送すれば良いのかを判断するのは難しく、リクエストの場合と同様に到達確率は低い。以上より、メッセージ転送の目的を持った端末を仮定しないで広域なオフライン地域内を通信するのは実用的ではないと言える。

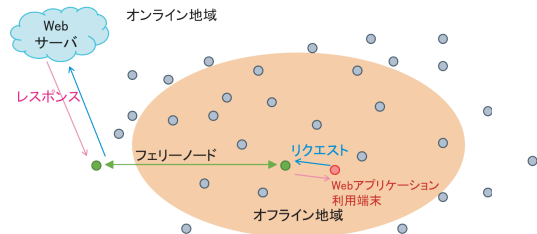


図 1: メッセージフェリー方式を用いた通信モデル

一方で、現実の災害においてメッセージ転送の目的を持った端末の存在を仮定するのは現実的であると思われる。具体的には、パケット中継を専門に行うトラックを被災地に巡回させられる可能性があり、また被災地をまわる物資輸送トラックや自衛隊の車両などにメッセージ中継のための機器を載せるという方法も考えられる。そこで本研究では図1に示すメッセージフェリー方式のルーティングを採用することを提案する。メッセージフェリー方式は、ある経路を計画的に移動する端末 (フェリーノード) を利用して効率的なデータ転送を行う方式の DTN モデルである [4]。この方式を用いればデータ転送は1ホップで済むのでメッセージ到達確率は飛躍的に高まる。

2.2 DTN フレームワークを用いた Web アプリケーション

広域災害時に利用可能な Web アプリケーションを実現する最も単純な方法は、DTN2[5] など既存の DTN フレームワーク上で既存の Web アプリケーションを利用することである。しかし、以下の二点からそれは実用的ではないと考えられる。

第一の理由は、通信データ量の制限である。平常時には通信データ量が比較的大きくても問題にならない場合が多いが、災害時には DTN ネットワーク資源は限定的であることから、通信データ量を意識した Web アプリケーションの実装が必要であると考えられる。

第二の理由は、Web アプリケーションの基本となっているリクエスト・レスポンス型の通信モデルにある。HTTP に代表されるように、通常の Web アプリケーションでは、クライアントのリクエストに対してサーバがレスポンスを返すことによって通信が成立する。クライアントはレスポンスが一定時間返ってこないときにはリクエストがレスポンスが失われたと判断してリクエストの再送を行うが、災害時に DTN フレームワークを用いると遅延の大きさが全く予想できないため盲目的に多くのリクエストが送られると考えられる。すると、リクエストのみならずそれに対応してレスポンスも大量に発行されてしまい、限りある DTN ネットワーク資源を無駄に消費する。つまり、災害時に Web アプリケーションを動作させる場合、平常時と同一のリクエスト・レスポンス型の通信モデルで動作させるのは効率的ではないと考えられる。

以上の理由から、平常時の Web アプリケーションを、そのまま既存の DTN フレームワーク上で動作させるだ

A Proposal and Implementation of a Disaster-tolerant Application Framework based on Message Ferry Routing
 Yuka Hasegawa[†] and Masato Oguchi[†]
[†]Ochanomizu University

けでは効率的な災害時通信は実現できないと考えられる。

3. 提案手法

広域災害時に実用的な Web アプリケーションを設計するための本研究での提案手法は次の二点にまとめられる。

- メッセージ転送の目的を持ったフェリーノードの存在を仮定し、端末・フェリーノード間で通信を行う方式を採用する。
- リクエストとレスポンスを分離した新たな通信プロトコルに基づく DTN フレームワークを提案する。

4. Web アプリケーションに提供する災害時 API

本章では、メッセージフェリーを用いたモデルの通信を実現する、災害時に利用可能な Web アプリケーションに提供する API の詳細を説明する。

Web アプリケーション実装の際は、端末側とクラウド側に分けて機能を実装することになる。提案 DTN フレームワークを用いることによってフェリーノードの存在を意識する必要はない。図 2 に提供される API とアプリケーションの関係を示す。

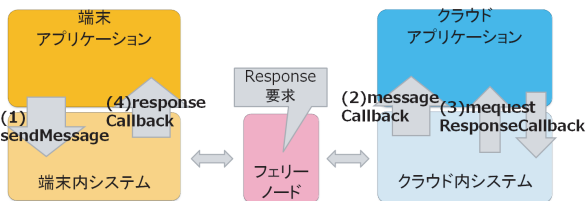


図 2: 提供される API とアプリケーションの関係

まず端末からクラウド方向の通信について説明する。端末アプリケーションが送信したいメッセージがある場合、(1)sendMessage を呼び出す。すると提案フレームワークがデータ通信を行い、クラウドアプリケーションで (2)messageCallback が呼び出される。次にクラウドから端末方向の通信について説明する。ここで、Response を要求するのは端末ではなくフェリーノードである。フェリーノードの Response 要求をトリガにクラウドアプリケーションで (3)requestResponseCallback が呼び出される。クラウドアプリケーションは指定された端末に対する Response を生成する。データ通信が行われ、端末アプリケーションで (4)responseCallback が呼び出される。

5. 災害時通信プロトコル

前章で説明した API を実現するための通信プロトコルについて、本論文ではリクエストとレスポンスの分離を実現する部分に絞って説明する。概要を図 3 に示す。

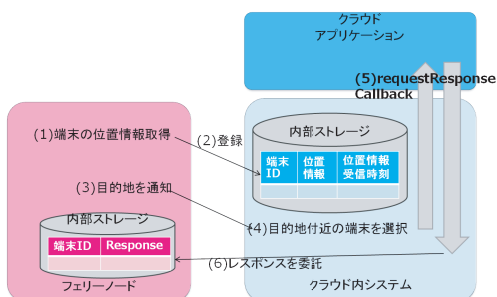


図 3: リクエストとレスポンスを分離するための手法

フェリーノードは端末と通信する際に現在地を取得しておき (1)、クラウド内システムに端末の現在地を登録する

(2)。フェリーノードがオフライン地域に出かける際に目的地を通知する (3)。クラウド内システムは各端末の位置情報を管理しているので、目的地周辺の端末へを選択し (4)、それらの端末への Response を requestResponseCallback を用いてクラウドアプリケーションに要求する (5)。生成された Response はフェリーノードに託される (6)。

6. 実機実装

今回、提案フレームワークを使用したアプリケーションの例として伝言アプリケーションを実装した。図 4 に伝言アプリケーションの Android 端末画面を示す。伝言の入力、宛先選択、送信ボタンまた受信した伝言の表示の機能がある。これを Google App Engine 上に実装したクラウドアプリケーションと接続して、動作を確認した。

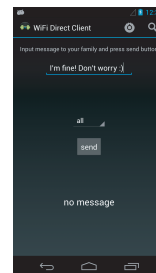


図 4: 伝言アプリケーションの Android 端末画面

7. まとめと今後の課題

大規模災害時にネットワークインフラが使用できなくなる地域において Web アプリケーションを実装するための DTN 通信モデルと Web アプリケーションの実装方針を提案した。大規模災害時にはメッセージフェリーの存在を仮定できることや従来のリクエスト・レスポンス型とは異なるモデルのアプリケーション実装が必要であることを論じた。アプリケーションに対して API を提供する DTN フレームワークを設計し、Android 端末と汎用クラウドを用いて実機実装を行った。

今後の課題として、メッセージフェリー型通信が適応できる環境をシミュレーションにより確認したい。また、フェリーノードの種類に応じて委託する Response の複製数や種類を変更するなどの検討を行いたい。

参考文献

- [1] Kevin Fall. "A delay-tolerant network architecture for challenged internets." Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp.27-34, ACM, 2003.
- [2] Amin Vahdat, et al. "Epidemic routing for partially connected ad hoc networks." Technical Report CS-200006, pp.18, Duke University, 2000.
- [3] 小山由, et al. "大規模災害時の安否確認システムと広域無線網利用可能エリアへの DTN に基づいたメッセージ中継法." 電子情報通信学会技術研究報告. Mo-MuC, モバイルマルチメディア通信 112.44, pp.171-177, 2012.
- [4] Wenrui Zhao, et al. "A message ferrying approach for data delivery in sparse mobile ad hoc networks." Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, pp.187-198, ACM, 2004.
- [5] DTN2, <http://www.dtnrg.org>