

# 大規模災害時における 各避難所間の情報共有システム構築に関する一検討

周爽<sup>1</sup> 小口正人<sup>1</sup>

**概要:** 近年、日本各地で地震、豪雨、台風、津波などの自然災害が連続して発生している。災害が発生する時に、家屋の倒壊などによって、自宅で生活できない人や、避難勧告、避難指示が発せられた人は避難所に行かなければならない。避難所は電気、通信、また情報など、様々な面で大災害時に頼りになる場所である。安全、安心の場として、避難者の避難所での生活を安全に過ごせるようにすることが必要である。避難所の物資などの情報共有を早急に行えることが望ましいが、東日本大震災や台風 19 号のような大規模災害により、インターネットが切れる可能性があるため、通信ができなくなる場合がある。その結果、避難所の物資とニーズを把握できなくなり、避難所に本当に必要な物資が届けられない状況になる。また、家族や友人の安否確認や避難所情報等を受け取ることができなくなることもある。そこで各避難所に置いておくエッジサーバを利用し、エッジサーバ同士が情報を同期しておくことで情報共有を可能にするシステムを提案する。また、個人情報への不正なアクセス、個人情報の紛失、破壊、改ざんおよび漏洩等のリスクに対しては、個人情報の適切な取り扱い、ならびに安全管理に取り組むことを検討する。

## A Study on the Construction of Information Sharing System between Shelters in A Large-Scale Disaster

SHUANG ZHOU<sup>1</sup> MASATO OGUCHI<sup>1</sup>

### 1. はじめに

近年、日本各地で地震、豪雨、台風、津波などの自然災害が連続して発生している。国土面積は世界の 0.25% に過ぎないが、マグニチュード 6 以上の地震の発生回数の割合は 20.8% と極めて多いものとなっている[1]。また、南海トラフ巨大地震や首都直下地震などの大地震が予想されている。南海トラフ巨大地震により、住宅全壊及び焼失棟数は約 95.4 万棟～約 238.2 万棟と想定されている[2]。

災害が発生する前に、避難勧告、避難指示が発せられた人や、家屋の倒壊、水や食料など物資の不足によって、自宅では生活できない人は避難所に行かなければならない。避難所は電気、通信、水、食料、情報など、様々な面で大災害時に頼りになる場所である。そのため、安全、安心の場として、避難者の避難所での生活を安全に過ごせるようにすることが必要である。

各避難所間の避難者の安否情報や避難所の物資などの情報共有を早急に行えることが望ましいが、東日本大震災や台風 19 号などのような大規模災害により、ネットワークインフラに大きな被害が発生する可能性が高い。津波の被

害により、ビル内の通信設備が損壊するなどの被害が発生したことがあった[3]。インターネットが切れる可能性があるため、大きな伝送遅延が発生したり、通信ができなくなってしまったりする恐れがある。それにより、避難所の物資とニーズを把握できなくなり、避難者が本当に必要な物資を手に入れることができず、必要ない物資の多くが残されてしまう混乱状態になる可能性がある。また、家族や友人の安否確認や避難所情報等を受け取ることができなくなることもある。

そこで、本研究では、大規模災害が発生する時に、劣悪なインターネット環境においても各避難所間の情報を共有できるシステムの構築を検討する。インターネットが途絶えた時に、各避難所に置いておくエッジサーバを利用し、エッジサーバ同士が情報を同期しておくことで情報共有を可能にする環境構築を検討する。DTN (Delay Tolerant Networking) 技術を用いることで、エッジサーバ間でデータの転送が実行できる。その環境を前提として、避難者、避難所や避難物資などの情報を収集し、各避難所間の情報共有システムの構築を提案する。また、個人情報への不正なアクセス、個人情報の紛失、破壊、改ざんおよび漏洩等のリスクに対しては、個人情報の適切な取り扱い、ならびに安全管理に取り組むことを検討する。

<sup>1</sup> お茶の水女子大学  
〒112-8640 東京都文京区大塚 2-1-1

## 2. 研究背景

### 2.1 関連技術

#### 2.1.1 DTN

DTN は、メッセージ指向のオーバーレイネットワークアーキテクチャであり、断続的な接続や大きな伝送遅延特性を持つネットワーク相互接続、異種ネットワーク相互接続の問題を処理することができる。2003 年、Fall 氏が惑星間インターネット問題を解決するために、「遅延/中断耐性ネットワーク(DTN)」という新しいアーキテクチャを提案した[4]。DTN の基本的な設計目標は、断続的な接続、高遅延、高ビットレートなどを特徴とする異種ネットワークの相互接続と、運用性をサポートすることである。DTN の概念が提案されたことで、深宇宙探査、潜水艦ネットワーク、軍事戦術ネットワーク、アドホックネットワーク、センサーネットワークなどの制約のあるネットワークの分野で一定の研究成果が得られた[5]。

DTN では、従来の TCP/IP 技術を拡張して、図 1 のようにバンドル層 (Bundle layer)、吸収層 (Convergence layer) と呼ばれる新しいアプリケーション層が実装されている。特徴技術は「中継転送技術」である[6]。

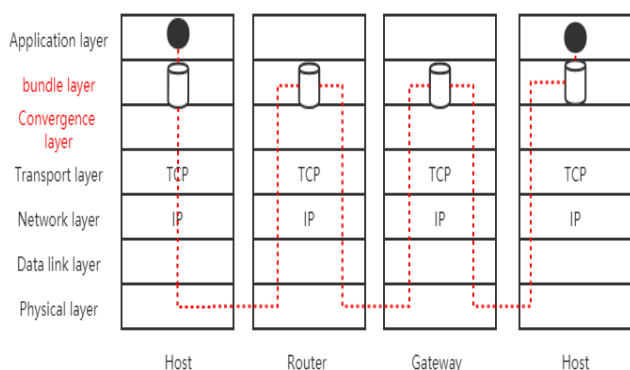


図 1 ネットワーク階層におけるバンドル層

アプリケーション層で作成したメッセージデータをバンドルに変換し、永続ストレージでバンドルを長期間保存する。通信可能な距離に次のホップノードが存在する場合、永久ストレージからバンドルを取り出して、次のホップノードに転送する。バンドル層を追加することで、通信ノードは頻繁に発生するネットワーク接続の中断や高遅延の影響を効果的に克服する。また、吸収層が設けられているため、今のインターネットや他の通信体系のネットワークがある場合にもそれらを繋ぎ、情報を伝達することも出来る。

#### 2.1.2 ウェブアプリケーション

ウェブアプリケーション (Web application) は、ウェブ (World Wide Web) 技術を基盤としたアプリケーションソフトウェアである。ウェブアプリケーションは、複数の静

的な Web リソースと動的な Web リソースから構成されている。

B/S (Browser/Server) モードは、インターネット技術の発展に伴い、C/S (Client/Server) モードを効果的に改良したものである。ブラウザを介したユーザインタフェースで実現されている。B/S モードのデータアクセスは、プレゼンテーション層からアプリケーション層、そしてデータ層までとする。また、B/S モードのシステムアーキテクチャは図 2 に示すように、プレゼンテーション層 (Presentation layer)、アプリケーション層 (Application layer)、データ層 (Data layer) に分かれる[7]。

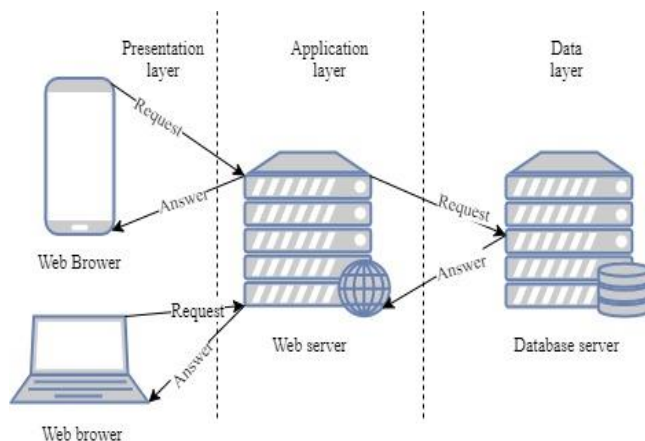


図 2 B/S システムアーキテクチャ図

ウェブアプリケーションはダウンロードしてインストールする必要がなくなり、ウェブブラウザ (Web browser) が搭載されているデバイスからインターネット環境に接続するだけでシステムを利用可能になる。

本研究の目的としては、避難所の状況管理が混乱している時に、脆弱な通信網の元でも確実に各避難所の状況を把握することである。そのため、B/S モードをベースにするウェブアプリケーションを提案する。災害の時に、時間がかからないのみならず、お年寄りでも簡単に使える必要がある。また、ウェブアプリケーションの論理的なトランザクションの大部分はサーバで実現されている。これにより、クライアントコンピュータやスマートフォンなどのマシンの負荷が軽減され、システムのメンテナンスやアップグレードの効率が向上される[8]。

### 2.2 関連研究

現在、災害発生時の通信障害の問題に対する DTN を用いたシステムが多く開発されている。

[9]では、災害が発生し通信インフラが機能しない状況で、避難所をクラスタと想定した。クラスタ間をフェリーノードが移動しており、クラスタ内の移動端末はフェリーノードにデータを集める。

[10]では、通信インフラが機能しない災害時初期の状況

において、災害時に動ける被災者が避難所に立ち寄ったり避難所から自宅へ戻ったりする際にも情報伝達を担う DTN ノードを検討した。動く被災者もキャリアと同様に、安否情報等を運搬し、キャリアと動かない被災者間の中継も行う。最新の災害情報を、避難所が広報として動く被災者やキャリアに運搬させ、動かない被災者に伝えさせる。

[11]では、災害発生後、DTN Server を搭載した車が被災地域を巡回する。DTN Server はインターネットに依存しない形で災害情報の登録、提供を行う。その後、再度被災地域を巡回し、それぞれのサーバに備蓄された情報を車載の DTN Server へ送信する。最後に、車載の DTN Server が集める情報をクラウドサーバへ送信し、災害情報の同期を行う。

### 3. 情報共有システムの利用形態

通常の状態では、図 3 に示すように、各避難所に平時から設置されているエッジサーバは、インターネットに接続されて動作するようになっている。

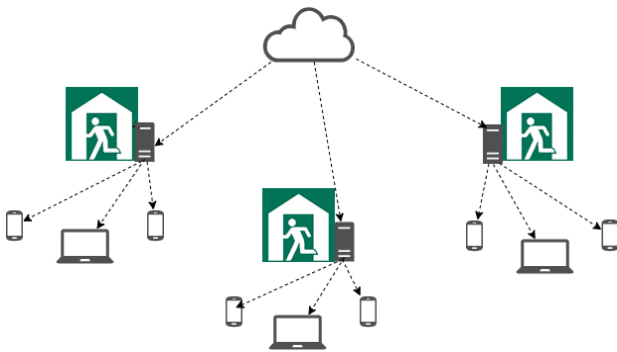


図 3 通常時の状況

大規模災害が発生する時に、各避難所に設置されているエッジサーバは、Wi-Fi で通信して互いにリンクを生成しデータをアップデートして、共有する。但し、インターネットが不安定な状態になった場合、極端に長い通信遅延時間、頻繁なパケット損失、間欠的な通信リンクの出現、などが発生する可能性がある場合、DTN 技術を用いることで、エッジサーバ間でデータの転送を実行する。

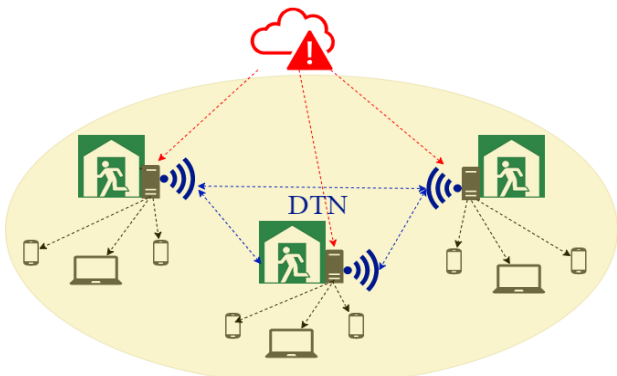


図 4 災害時の状況

各避難所間の情報共有システムはそのような環境を前提として、提案される。

## 4. 実験環境

### 4.1 Node.js

本研究では、Node.js を採用する。

Node.js[9]は Chrome V8 エンジンベースの JavaScript 実行環境である。Node.js は、PHP、Python、Perl、Ruby などのサーバサイド言語と同等のスクリプト言語として、JavaScript を動作させることができるサーバサイド開発プラットフォームである。2009 年 5 月に Ryan Dahl により開発された。

Node.js は二つの特徴がある。

- ・ ノンブロッキング I/O (non-blocking I/O)

Node.js は、JavaScript と同じく、シングルスレッドで動作する。一度に 1 つの処理しかできないため複数の処理を捌けない。そのため、Node.js ではノンブロッキング I/O を採用している。データの送受信の完了を待たず、それぞれ並列処理が可能になる。これにより大量なアクセスにも耐え、高速処理を実現している。

- ・ イベントループ (Event Loop)

イベント駆動モデルでは、各 IO ジョブがイベントキューに追加され、スレッドはキュー上のタスクをループする。実行中にブロック (ファイルの読み込み・データベースへの問い合わせ) がある場合には、スレッドは停止して結果を待つのではなく、結果を処理するためにコールバック関数を残し、キュー内の次のタスクを処理する。キューに渡されたコールバック関数は、ブロッキングタスクが実行された後に、スレッドからのみ呼び出される (図 5)。

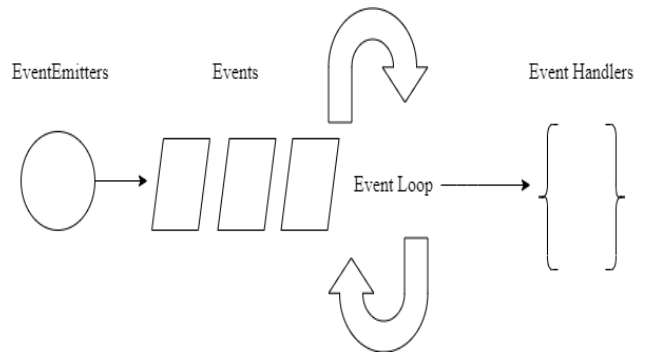


図 5 イベント駆動モデル

### 4.2 Express Platform

本研究では、Express フレームワークを採用する。

Express フレームワークは柔軟性が高い Node.js の Web アプリケーションフレームワークである。様々な Web アプリケーションの作成に役立つ強力な機能と豊富な HTTP ツールを提供している。Express を使えば、完全に機能的なウェブサイトを素早く構築することができる。また、Express

フレームワークをベースにして、MVC (Model, View, Controller) アーキテクチャを構成できる。

モデル (Model) は、アプリケーションのデータロジックを処理するために、使用される。ビュー (View) とは、アプリケーションの中で、データの表示を処理するものであり、モデルデータを元に作成される。コントローラ (Controller) は、ユーザのインタラクションを処理するアプリケーションの部分である。ビューからデータを読み取ったり、ユーザの入力を制御したり、モデルにデータを送信したりする役割を担っている。

MVC の動作を、図 6 に示す。

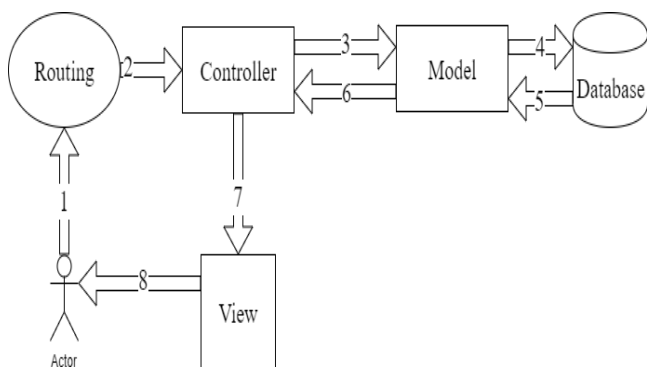


図 6 MVC

1. ブラウザはユーザからのリクエストを受け、URL をルーティングに送る。
2. ルーティングは、URL によって、どのコントローラを呼び出すかを判断する。
3. 呼び出されたコントローラはモデルと打合せする。
4. モデルはデータロジックを処理した結果をデータベースに反映する。
5. モデルはデータベースからのデータを受け取る。
6. 受け取ったデータをコントローラに渡す。
7. コントローラは受け取ったデータをビューに引き渡す。
8. ブラウザに Web ページが表示される

Express フレームワークは三つのコア機能を持っている。

- HTTP リクエストに応答するように、ミドルウェアを設定することができる。
- さまざまな HTTP リクエストアクションを実行するためのルーティングテーブルを定義することができる。
- パラメータをテンプレートに渡すことで、動的に HTML ページをレンダリングすることができる。

#### 4.3 MongoDB

本研究では、MongoDB を採用する。

MongoDB は NoSQL のドキュメント指向データベースである [10]。これらのドキュメントは、JSON (JavaScript Object Notation) 形式で MongoDB に保存される。大量のデータ

を扱うシステムにおいて、リレーショナルデータベースに比べ、MongoDB は比較的簡単にスケールできるような仕組みを持っていることが特徴的である。

災害が発生し、アクセスが多い場合、MongoDB はデータの追加、更新、削除、クエリを高速に行うことで、効率的な対応ができる。

## 5. システム概要と課題

### 5.1 システム機能

本研究は、避難者、避難所や避難物資などの情報を収集し、各避難所間の情報共有システムの構築を検討する。本システムには三つのロールがある。User は普通の避難者である。Admin は避難所の管理者である。Root はシステム管理者である。

避難者は、このシステムを使うことで、自分の情報の登録、変更ができる。また、家族や友人の安否情報を友達追加で、確認することができる。避難所の管理者は、避難所の情報、備蓄物資や需要物資などの登録、変更、削除ができる。システム管理者は、システムの権限管理、ロール管理、体調管理ができる。

情報共有システム機能は図 7 に示すように、四つのモジュールに分かれている。

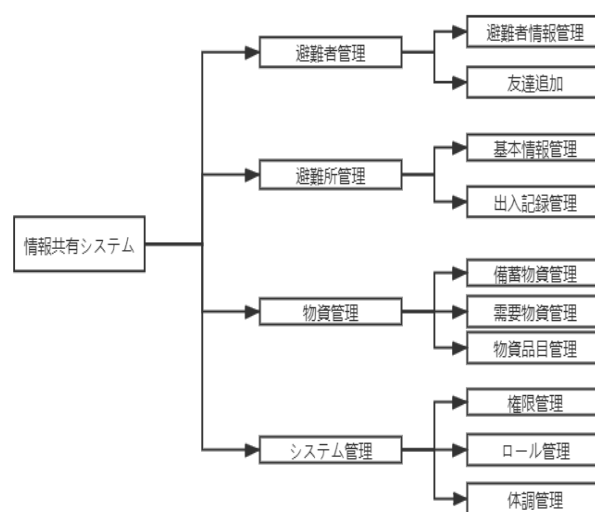


図 6 システム機能

#### 1. 避難者管理モジュール

このモジュールでは、避難者情報の登録、照会、更新、削除することができる。避難者同士は、友達を追加することで、お互いの状況を確認することができるようになる。

#### 2. 避難所管理モジュール

このモジュールでは、避難所の基本情報の照会、登録、更新、削除ができる。また、避難所にいる避難者の出入記録の照会、登録、更新、削除ができる。この機能によって、

各避難所の管理者は避難者の状況を把握できる。

### 3. 物資管理モジュール

このモジュールでは、避難所の管理者が備蓄物資管理によって、避難所の備蓄物資の品目と数量を更新、削除することができる。また、需要物資管理によって、避難所の需要物資の品目と数量をシステムに登録することができる。備蓄物資と需要物資の品目がない場合には、物資品目管理で新しい品目を登録し、所属避難所を選択し、数量を登録することができる。

### 4. システム管理モジュール

このモジュールでは、システムの権限管理、ロール管理、体調管理ができる。その三つの権限はシステム管理者だけが持っている。

## 5.2 課題

このシステムでは個人情報の登録が必要ため、個人情報への不正なアクセス、個人情報の紛失、破壊、改ざんおよび漏洩等のリスクに対しては、個人情報の適切な取り扱い、ならびに安全管理に関する仕組みを考えなければならない。

システム管理者には、避難者の個人情報データにアクセスする場合、アクセスログを記録することやデータにアクセスする前に許可をもらうことなどの制約があればセキュリティを大幅に強化することが可能である。

偽者としての不正行為を防ぐため、本人確認が必要である。但し、災害の時、身分証明書を避難所に持って来ないこともあるので、本人確認ができなくなる。そこで事前に登録済みのユーザ、すなわち本人確認ができていないユーザの情報と発災後システムに登録するユーザ、すなわち本人確認ができていないユーザの情報を分けて保存することで、セキュリティを強化することができると思われる。

友達を追加する時には、本人確認ができていないユーザ同士の場合、電話番号だけで友達を追加できる。本人確認ができていないユーザの場合、付加的な確認をする仕組みがあれば良い。また、情報を共有する時、個人情報の安全確保のため、家族と友人とは共有できる情報が異なるようにする必要がある。

## 5.3 データベース

システムのデータベースは図7に示すように考えられる。

データベースはユーザ (user)、避難所 (shelter)、避難物資 (supplies)、ロール (role)、健康状態 (status) という五つの実体がある。ユーザは、一つの健康状態、ロールを持ち、一つの避難所に所属する。また、ユーザ同士は、友達関係がある。避難所と物資は、避難所の備蓄物資と避難所の需要物資という二つの関係がある。

ユーザはユーザ ID、名前、パスワード、性別、生年月日、年齢、住所、電話番号、メモの属性を持っている。

避難所は避難者 ID、避難所の名前、アドレス、受容人数と滞在人数の属性を持っている。

避難物資は避難物資 ID と名前の属性を持っている。

健康状態は健康状態 ID と状態の属性を持っている。状態は無傷、軽傷、重傷、重体、死亡を設定する。

ロールはロール ID とロール名前の属性を持っている。避難者、避難所管理者、システム管理者を設定する。

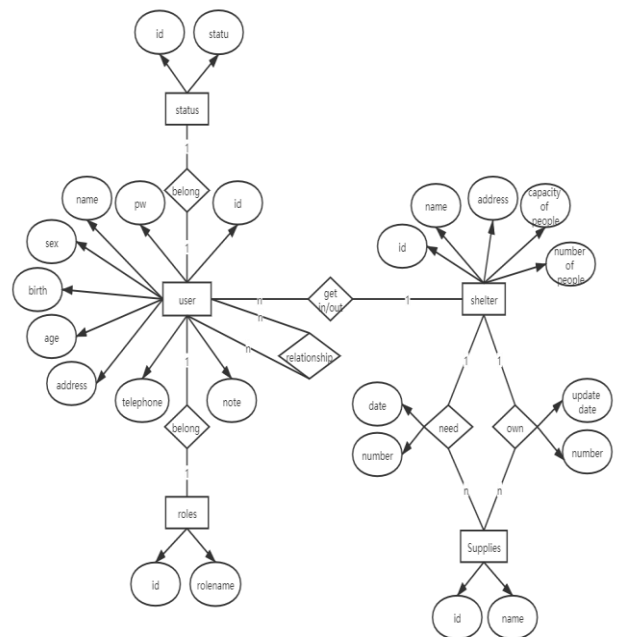


図7 実体関連図

## 5.4 システム実装

本情報共有システムの避難所基本情報管理機能を例として紹介する。フローチャートを図8に示す。

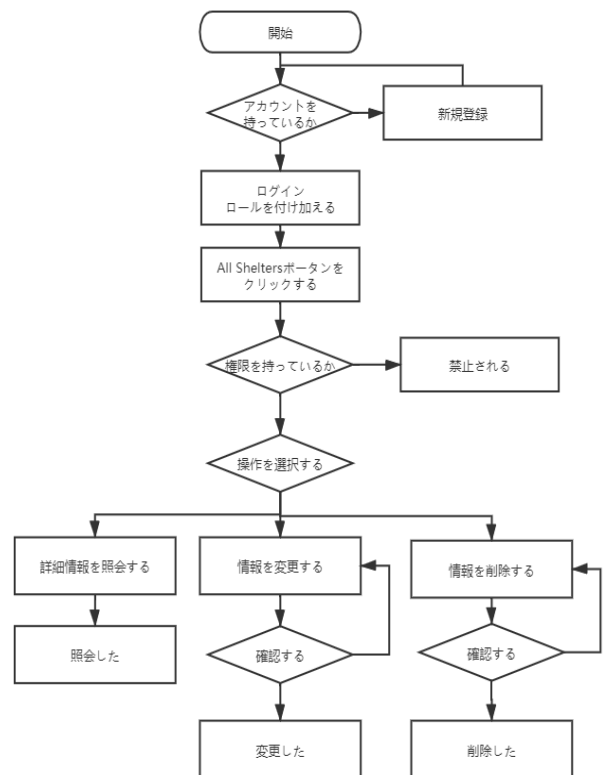


図8 フローチャート

最初は登録のページが表示される。アカウントを持っている場合は、電話番号とパスワードを入力し、ログインする。アカウントを持っていない場合は、新規登録ボタンを押し、新規登録ページで登録する。

ログインが成功した時には、Node.js の ACL (Access Control Lists) を使って、ユーザにロールの権限を付け加える。ログインして、システムのホームページが表示される。ユーザは All Shelters というボタンを押し、ルータは ACL を使って、URL による権限を持っているかを判断する。権限を持っていない場合は、リクエストが禁止される (図 9)。権限を持っている場合は、すべての避難所基本情報管理のページが表示される (図 10)。

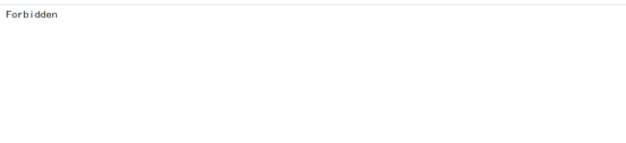


図 9 権限を持っていない

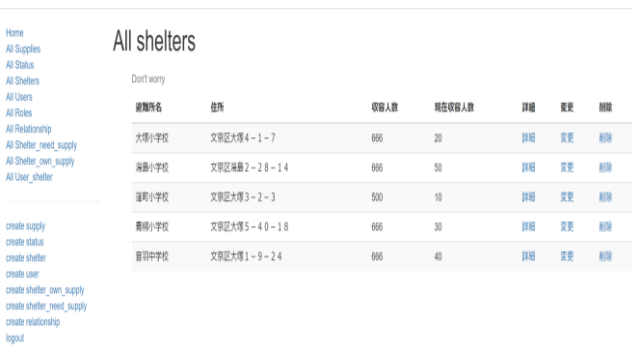


図 10 避難所基本情報管理ページ

ユーザは詳細、変更、削除のボタンを押すと、それぞれのページが表示される。確認ボタンを押すと、避難所基本情報の管理ができる (図 11)。



図 11 避難所基本情報の変更

## 6. まとめと今後の課題

大規模災害が発生する時に、大きな伝送遅延が発生する可能性や、通信障害が発生する恐れがある。そこで本研究では、各避難で置いておくエッジサーバを利用し、DTN 技術を使う環境を前提として、情報共有できるウェブアプリケーションを提案した。今後は情報共有ウェブアプリケーションの実装を進め、分散される環境で実証実験を行う予定である。また、分散されたエッジサーバ間においてデータを同期する方法を考察する必要がある。

## 謝辞

本研究を進めるにあたり、UCLA の高井峰生先生および情報通信研究機構の大和田泰伯様から有用なアドバイスを頂きました。また、本研究は一部、JST CREST JPMJCR1503 の支援を受けたものです。ここに感謝の意を表します。

## 参考文献

- [1] 内閣府：防災情報のページ，“防災白書”，平成 18 年
- [2] 内閣府：防災情報のページ，“南海トラフの巨大地震被害想定（第一次報告および第二次報告概要）”，平成 25 年度
- [3] 総務省：“情報通信白書”，平成 25 年版
- [4] 叶建设, 宋世杰, 沈荣骏. 深空通信 DTN 应用研究[J]. 宇航学报, Vol.031, No.004, p.941-949, 2010
- [5] Fall K. “A Delay-Tolerant Network Architecture for Challenged Internets” Applications, Technologies, Architectures, and Protocols for Computer Communications, Aug., 27-34, 2003
- [6] 鶴正人, et al. “DTN 技術の現状と展望” 通信ソサイエティマガジン, No.16, p.57-68, 2011
- [7] Lidan Mao. “Application of Browser/Server Architecture in College English Online Learning System Design”, Vol.13, No.03, p.129-140, 2018
- [8] Kyung-Hwan Ahn, Sung-Kwang Kim, Kwan-Pyo Hong and Ki-Jun Han. “Design and implementation of browser/server environment-based hospital information search system (BS-HISS)”, IEEE, Vol.2, p.1569-1572, 1999
- [9] 金田知展, 中村嘉隆, 高橋修. “DTN を用いた災害時通信システム構築法の提案”, マルチメディア, 分散協調とモバイルシンポジウム 2013 論文集, p. 964-969, 2013
- [10] 山根学, 小花貞夫. “DTN を用いた災害時初期の情報伝達方式に関する一考察” 第 76 回全国大会講演論文集 2014.1, p.483-484, 2014
- [11] 菊池瑤介, 内田法彦, 柴田義孝. “大規模災害を考慮したクラウド型災害情報共有システムの構築” 第 75 回全国大会講演論文集 2013.1, p.813-814, 2013
- [12] Node.js 入手先 <https://nodejs.org/zh-cn/>
- [13] Madison, Michael. “NoSQL Database Technologies”, No. 6150282, 2015.