

6U-01

# 災害時の DTN 対応安否確認アプリケーションの設計に関する一検討

高田 千暁<sup>†</sup>

黒崎 裕子<sup>†</sup>

高井 峰生<sup>†‡</sup>

小口 正人<sup>†</sup>

<sup>†</sup>お茶の水女子大学

<sup>‡</sup>UCLA

## 1. はじめに

今やインターネットは様々なアクセス網を包含し世界中を結ぶことのできる情報交換・共有システムとして社会・経済のインフラともいえる役割をはたしており、社会はネットワークに依存したものとなっている。それに伴い、従来は想定されていなかったような劣悪な環境下での通信アプリケーションの要求がでてきた。中でも、地震などの災害によって本来の通信インフラ機能や性能が低下、停止した場合の緊急代替情報交換手段を事前に設計、準備しておくことは防災や減災を考える上で大変重要である。現在、災害時には各キャリアや日本限定で Facebook から安否確認掲示板が提供されたり、goo アプリなど地震や災害に備えたアプリが用意されているが、そういったサービスはインターネットが機能しているという前提で考えられている。

そこで本研究では、地震などによって地域的にインターネットが機能しないような劣悪な条件下でも部分的に稼働しているサーバ機能付 Wi-Fi アクセスポイントを用い、Delay/Disruption Tolerant Network (DTN) 技術を利用して安否確認が行える通信アプリケーションの開発とその実装要件を検討する。

## 2. DTN

DTN は遅延耐性ネットワークとも呼ばれ、物理的なリンクの切断やデータの送受信遅延に対応していない TCP/IP 技術を拡張させた「中継転送技術」である [1]。この一般的な手法は「届きそうな」端末にデータを送信し、端末間でデータをホップさせていくことで目的端末まで届かせるというものである。これにより、中断や切断が多発したり、極端に長い通信遅延が生じたりするような劣悪な通信環境下でも信頼性のあるデータ転送を実現することができる。

## 3. 災害時の想定環境

災害時のネットワーク環境として、サーバ機能付 Wi-Fi アクセスポイントを用いた無線メッシュ/DTN システム (図 1) を想定している。このシステムは、使えなくなった経路が発生しても稼働している一部のサーバ機能付 Wi-Fi アクセスポイントと DTN 技術を用いて継続的に接続、再構成を繰り返し、送信先に達するまでノードからノードへ転送を行って目的端末まで到達させることを可能とするものである。これにより、大規模災害などによって地域的にネットワーク接続が切れている場合でも通信が可能となる。本研究ではこのような環境を想定して検討していく。

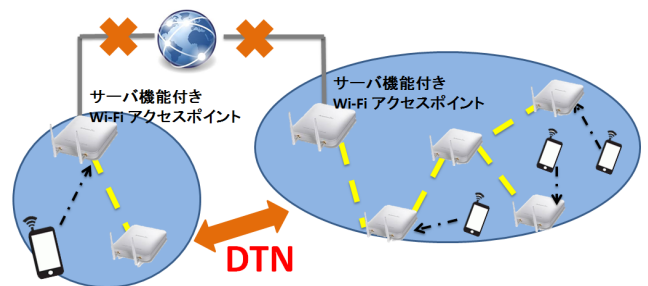


図 1: サーバ機能付 Wi-Fi アクセスポイントを用いた無線メッシュ/DTN システム

## 4. 災害時アプリケーションの検討

### 4.1 アプリケーション概要

本研究では Android 端末を用い、稼働しているサーバ機能付 Wi-Fi アクセスポイント同士のデータ同期を可能とするような Android アプリケーション (以下、アプリ) の設計を行った (図 2)。

- 1) アプリは災害が起きネットワーク環境が不安定になった後でもインストールできるように、アクセスポイントに保存されている。ユーザはここからアプリをインストールする。
- 2) アプリを用いて安否情報を記入し、アクセスポイント 1 (以下、AP1) に送る。このときユーザは、自身のデータを送信すると同時に AP1 に保存されているデータを受け取り、端末に保存する。
- 3) ユーザが動くことで端末がアクセスポイント 2 (以下、AP2) とデータのやり取りを行い、非接続状態の AP1 と AP2 のデータが同期される。

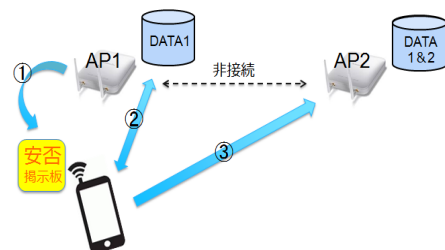


図 2: アプリケーション概要

### 4.2 アプリ設計における注意点

アプリを検討するにあたり、注意しておかなければならないこととして各ポイントでのデータ取得の時間のずれがあげられる。例えば、端末から AP1 には PM2:00 にデー

A Study on Design of Safety Confirmation Application Corresponding with DTN in the Case of a Disaster

<sup>†</sup> Chiaki Takada, Yuko Kurosaki, Masato Oguchi

<sup>†‡</sup> Mineo Takai

Ochanomizu University (<sup>†</sup>)

UCLA (<sup>‡</sup>)

タが届き、インターネットには PM5:00 に届いたとする。このときデータの取得には 3 時間のずれがでているが、このずれに気づかずにインターネットにデータの問い合わせを PM3:00 にしてしまうと、データは「まだ届いていない」だけに関わらず、「ない」と勘違いされてしまう可能性がある。また、各アクセスポイントはつながっておらず同期ができていないので、インターネット側はどのアクセスポイントといつの時点のデータを同期したかを把握しておかなければならない。さもなければ、実際は古いデータにも関わらず届いた時間が新しいからといってそのデータを同期してしまうといったことが起こりうる。

アプリの設計はこのような問題に注意しながら検討していく。

#### 4.3 端末側の実装

Google が提供している Android 開発環境である Android Studio で実装した。端末としては Galaxy Nexus, Android 4.2.2 を用いた。端末側アプリの主な動作は以下の通りである。

- 1) ユーザは安否情報を記入し、端末が保存しているデータに追加する。初めて使用する場合は登録したデータのみが保存される形となる。
- 2) アクセスポイントに端末が保持するデータのリストを送り、サーバのデータとの差分を調べる。
- 3) 端末のみが保持するデータをサーバへ送り、サーバから取得したデータを保存する。

ユーザが記入するものは、名前、電話番号、被害状況、居場所、コメントの 5 つであり、データの送信と同時にデータを送った時間と通信先のサーバの ID が自動的に取得される。ただし、災害が起きた場合でも動作している機器が保有する時刻に大幅なずれは生じないということを前提としている。また、各データには ID がつけられ、これにより保存データのリストを作成している。

端末とサーバがデータのやり取りをする場合、データは JSONObject 型で扱われる。JSONObject とは {} で囲まれたものが 1 つのオブジェクトであり変数名と値のセットによって成り立っているが、これには 1 人分のデータしか入らない。よってデータは、JSONObject 型がリスト状になった JSONArray 型に複数人のデータを格納し、それを変数名 data の値とした JSONObject 型になっている。

#### 4.4 サーバ側の実装

サーバとしては、メモリ 3.7 GB の Intel Pentium E2220 2.4GHz を 2 台用いた。サーバは java で実装しており、主な動作は以下の通りである。

- 1) 端末から通信がきたらサーバの ID を返す。
- 2) 端末から送られたデータのリストをもとにお互いが保持していないデータを判断し、端末のみが保持するデータの送信を端末へ要求する。

- 3) 端末から送られてきたデータをデータベースに保存し、サーバのみが保持するデータを端末へ送信する。

端末側で述べたように端末とサーバでデータのやり取りを行う場合はデータは JSON 形式で扱われているが、サーバ内ではデータベースとして保存されている。よって、検索機能などをつけたい場合は端末側ではなくサーバ側で検索する形となる。

データの保存方法は update 文で上書きするのではなく、insert 文のみを用いて追加だけしていくようにしている。これは、例えばユーザのデータ送信履歴から行動軌跡をたどるなど、災害時においては過去のデータも価値がある可能性が考えられるためである。

#### 5. アプリの動作確認

端末からアクセスポイントまでは IEEE802.11n の無線 LAN で、アクセスポイントからサーバへは有線 LAN でつなぎ、2 台のサーバには重複を含む 10 個のデータが保存されている。この環境下で端末でアプリを起動させ、必要事項を記入した後、サーバ 1 とサーバ 2 へ順にデータを送信した。送信するサーバの切り替えはアプリのボタンで行う。

サーバ 1 にデータを送信した後、端末にはサーバ 1 のデータが保存された。次にサーバ 2 とデータのやり取りを行うと、端末に保存されているサーバ 1 のデータでサーバ 2 が保持していないデータのみがサーバ 2 へ送られた。それと同時に、端末は自身が保持していないサーバ 2 のデータを取得できたので、端末とサーバ 2 のデータが同期されたことが確認できた。この状態でさらにサーバ 1 とデータのやり取りを行えば、サーバ 1 とサーバ 2 は基本的に同期された状態になると考えられる。また、ユーザが安否情報を記入し送信ボタンを押すと、データの ID や送信時間、サーバの ID も取得できていることが確認できたので、アプリは正しく動作しているとわかった。

#### 6. まとめと今後の課題

地域的にインターネットが機能しないなど、従来は想定されていなかったような劣悪な条件下でも部分的に稼働しているサーバ機能付 Wi-Fi アクセスポイントと DTN 技術を利用して、安否確認が行えるようなアプリケーションの検討と簡単なアプリの実装を行った。2 台のサーバと 1 台の端末でアプリを起動させデータのやり取りを行った結果、非接続状態のアクセスポイントでもデータの同期ができそうだとことが確認できた。

今後の課題としては、アプリの動作確認に加え、その評価を行っていく。また、サーバや端末、データの数を増やしていきつつ、より効率のよい同期方法を検討していきたい。

#### 参考文献

- [1] 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[春号], pp.57-68, 2011.