災害時のDTN対応安否確認アプリケーションの実装と予備評価

高田　千暁†　　黒崎　裕子†　　高井　峰生††　　小口　正人†

† お茶の水女子大学　〒112-8610 東京都文京区大塚 2-1-1
†† UCLA Computer Science Department　3803 Boelter Hall, Los Angeles, CA 90095-1596, USA
E-mail: †{g1120526,g1020510}@is.ocha.ac.jp, ††mineo@ieee.org, †††oguchi@computer.org

あらまし　インターネットは情報交換・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，社会のネットワーク依存度はますます高まってきている．それに伴い，地震などの災害でその機能や性能が低下，停止した場合の緊急代替情報交換手段が必要となってきた．また，これらのことを事前に設計，準備しておくことは地震大国である日本にとって防災や減災を考える上でも大変重要になってくる．現在においても，アプリケーションや各キャリアからの安否掲示板など地震や災害に備えたサービスは存在するが，これらはインターネットが機能していることを前提としている．そこで本研究では，地域的にインターネットが機能しない劣悪な条件下でも，部分的に稼動しているサーバ機能付 Wi-Fi アクセスポイントと 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を開発しその実装と予備評価を行った．

キーワード　DTN，災害時アプリケーション

Implementation and Preliminary Evaluation of Safety Confirmation Application Corresponding with DTN in the Case of a Disaster

Chiaki TAKADA†, Yuko KUROSAKI†, Mineo TAKAI††, and Masato OGUCHI†

† Ochanomizu University　　2-1-1 Otsuka, Bunkyouku Tokyo 112-8610 JAPAN
†† UCLA Computer Science Department　3803 Boelter Hall, Los Angeles, CA 90095-1596, USA
E-mail: †{g1120526,g1020510}@is.ocha.ac.jp, ††mineo@ieee.org, †††oguchi@computer.org

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（以下，アプリと呼ぶ）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の機能や性能が低下，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も大変重要になってくる．現在，各キャリア [1]〜[3] や日本限定で Facebook [4] から災害発生時に利用可能となる安否確認掲示板が提供されていたり，goo 防災アプリ [5] など地震や災害に備えたアプリが用意されている．しかし，このような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，地震などの災害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動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アプリの開発とその実装の予備評価を行う．

以下，2 節では従来研究について，3 節では災害時に想定されるネットワーク環境を説明する．4 節では検討しているアプリを紹介し，5 節で実装したアプリの検証実験について述べ，6 節では従来手法との比較評価を行った．最後に，7 節でまとめと今後の課題について述べる．

2. 従来研究

2.1 関連研究

DTN 技術を用いた研究は数多く検討されている．

[6] ではバッテリの残量から転送先を動的に選択し，被災地

内の情報を被災地外へ運び出す災害時通信システム構築法の提案がなされている．この研究では広域災害時により通信インフラが機能しない場合に，避難所にいる被災者の安否情報や被災状況などの情報を収集し，メッセージフェリー方式を用いて被災地の外へ運び出すことを想定している．このとき避難所内の移動端末の通信では，バッテリ残量が多い端末へデータが集まるようにデータ転送先が動的に選択される．これにより，特定の移動端末に負荷が集中することを防ぎ，ネットワーク持続時間を長期化することを目指している．

[7] ではモバイルアドホックネットワーク (MANET) と DTN の異なる 2 つのネットワーク形成技術を高度に融合した端末間通信技術が提案されている．これは変化する周囲の状況にあわせて適切なネットワークモードを自動的に判断して切り替わることで，通常は圏外で通信不可能な環境であっても効率的な通信の実現と消費電力の低減を目指しており，全行程 2.7km での実証実験もなされている．

これらの研究はどちらもアクセスポイントを介さずに機器同士が直接通信を行うアドホックモードを用いて，情報のやりとりがなされている．転送先の動的な選択やネットワークモードの切り替えにより端末のバッテリ消費はある程度抑えられると予想できるが，これはまだ解決すべき問題である．また，すれ違いなど端末が近い位置にないと行えないこと，ホップすることによってスループットが低下してしまうといった課題もある．アドホックモードを用いる場合は，これらの課題を考慮していく必要があると考える．

2.2 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 [8]．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．代表的な技術として蓄積運搬形転送がある．

この技術は，まず宛先に届きそうなノードにデータを転送し，次に中継ノードにデータを蓄積する．この一時的な蓄積により，再開を待つ，あるいは最適なタイミングを待つことができ時間的不連続性に対応できる．さらに蓄積されたデータを物理的に，例えば自動車や列車を用いて運搬する．この運搬によって空間的不連続性に対応でき，また，運搬によって無線通信の距離を短くすることができるので大容量のデータ転送を無線資源を浪費することなく実行することが可能となる．そして宛先が見つかったら，運搬された保存データは転送される．

他にも， Epidemic Routing という手法がある．この手法は，ノードの移動により遭遇した全てのノードに情報を複製して伝達していくという情報伝達技術である．そのため，複製するデータ数は指数的に増えてしまうが，データ転送遅延は小さくすることができる．

既存の DTN フレームワークとしては IBR-DTN [9] などがある．

3. 災害時の想定環境

災害時の通信環境として，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．このシステムは災害が発生したときサーバ機能付 Wi-Fi アクセスポイントが通常時モードから非常時モードに切り替わることで，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．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を行う．

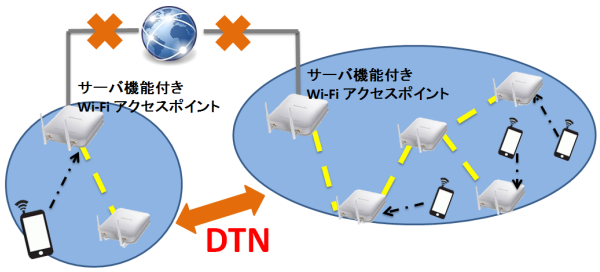

図 1　サーバ機能付 Wi-Fi アクセスポイントを用いた無線メッシュ/ DTN システム

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，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，そのデータをアクセスポイント 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．

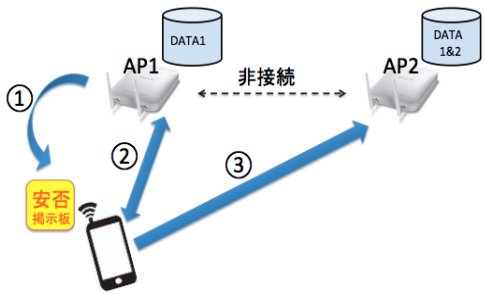

図 2　アプリ概要

4.2　アプリケーション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 にデータが届いたが，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 時間のずれが生じている．この時間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」だけにも関わらず,「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データの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た．

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．このとき，送信した時間，データの ID ，サーバの ID も自動的に取得されユーザ情報に含まれる．初めてアプリを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)　サーバから要求があった端末のみが保持するデータをサーバへ送り，サーバから送られ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 項目であり，データを送信すると同時にデータの ID ，送信した時間，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．データのリストは各データにつけられた ID から作成している．これをデータ本体のやり取りを行う前にサーバへ送信し，サーバと端末のデータの差分を調べることで，重複データの送受信を防いでいる．

端末とサーバがデータのやり取りを行う場合，データは JSON 形式にしており， 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．例えば, { ”name” : ”お茶の水花子” } ならば変数名が「name」，その値が「お茶の水花子」となる．しかし， JSONObject 型には 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ている 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 型としている．データはサーバとのやり取り以外では，ユーザ情報の各要素をもったクラス List⟨*BoardData*⟩ 型として端末に保存されている．データの ID は int 型，時間は long 型，その他の要素は String 型である．

4.4　サーバ側の実装

サーバとしては，メモリ 3.7GB の Intel Pentium E2220 2.4GHz を 2 台用いた．サーバは Java で実装してあり，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 JSONObject 型で扱われているが，サーバ内ではデータの ID を PRIMARY KEY としてデータベースで保存されている．よって，検索などを行いたい場合は端末側で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 文で上書き保存するのではなく，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.　検 証 実 験

サーバ 2 台（サーバ 1，サーバ 2）と端末 1 台を用いて，アプリの動作確認を行った．

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 でつないだ．サーバ 1 とサーバ 2 には各々10 件ずつデータが保存されており，そのうち 3 件のデータはサーバ 1 とサーバ 2 の両方に含まれる重複データとなっている．この環境下で端末でアプリを起動させユーザ情報を記入し，サーバ 1 へデータを送信する（①）．次にサーバ 2 にもデータを送信し（②），各送信後のサーバ 1，サーバ 2，端末の保存データ数を確認する．どのサーバとやり取りを行うかは IP アドレスで指定でき，現段階では固定 IP アドレスを保持しているサーバ 1 とサーバ 2 をアプリのボタンによって切り替えできるようにしている．実験の概要

を図 3 で示す．

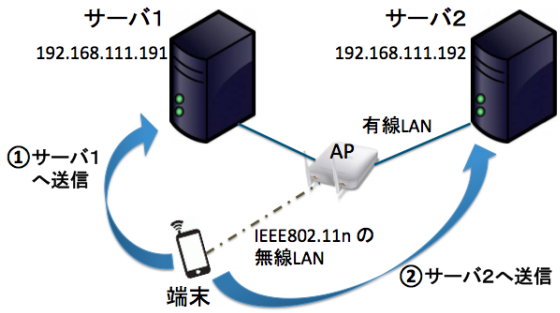

図 3　実験概要図

まず始めにサーバ 1 へデータの送信を行った．するとサーバ 1 の保存データ数は，端末から送られたデータ 1 件と元々保存されていた 10 件を合わせた 11 件となった．端末側を確認すると，サーバ 1 が保持していた 10 件のデータが送られるので，端末の保存データ数は 11 件となっていた．よって，送信後にサーバ 1 と端末のデータが同期されたことがわかった．また，端末から送られたデータの内容を確認すると，記入した情報以外にもデータの ID，送信した時間，サーバの ID が含まれていたので，送信したと同時に必要な情報の取得もできていることがわかった．次に，サーバ 2 にデータを送信した．サーバ 2 には，端末が保持している 12 件のデータ（サーバ 1 との送信で得た 11 件と新たに記入したユーザ情報 1 件）から重複している 3 件のデータを除いた 9 件が端末から送られるので，サーバ 2 の保存データ数は元々の 10 件とこの 9 件を合わせた 19 件となった．端末も同様で，サーバ 2 から重複データを除いた 7 件のデータが送られるので，端末の保存データ数は元々の 12 件と合わせて 19 件となり，サーバ 2 と端末のデータが同期されたことがわかった．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．

以上のことから，サーバ 2 台と端末 1 台ではアプリが正しく動作したことが確認できた．

6.　従来手法との比較評価

提案手法と既存研究で用いられている手法の大きな違いは，アクセスポイントを介してデータのやり取りを行うか否かである．既存研究ではアドホックモードを用いてデータのやり取りを行っているため，全ての端末同士にリンクが作られる．一方，提案手法では，アクセスポイント対アクセスポイント，アクセスポイント対端末の 2 通りでリンクが作られ，端末対端末には作られないので，2 つの手法では総リンク数が変わってくる．そこで，提案手法の場合と Epidemic Routing でアドホックモードを用いてデータのやり取りを行った場合では，どの程度総リンク数が変化するのかを比較した．

6.1　比較シナリオ

想定する環境として，50m × 50m の範囲内に端末がランダムに配置されており，これらの端末が移動しない状況を考えた．Epidemic Routing では端末の数を 50 台とし，提案手法では端末とアクセスポイントの合計数を 50 台とする．端末とアクセスポイントの通信距離はどちらも 10m として，端末やアクセスポイントを中心とした半径 10m の円の範囲内にノードが含まれているときにリンクは作られ，円の範囲外にノードがあるときはリンクが繋がっていないとする．このとき，提案手法のアクセスポイント数によってどの程度リンクの総数が変化するかを比較していく．

リンクの状態については図 4 で、想定環境については表 1 に示す．

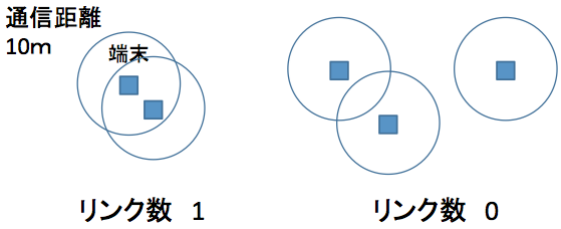

図 4　リンクの状態

表 1　想 定 環 境

想定範囲	50 m× 50 m
各ノードの通信距離	10 m
ノードの合計個数	50 台

6.2　比 較 評 価

提案手法のアクセスポイント数を 1 から 8 まで増やしたときの各場合において 5 回ずつ結果をとり，その平均を図 5 のグラフで示す．

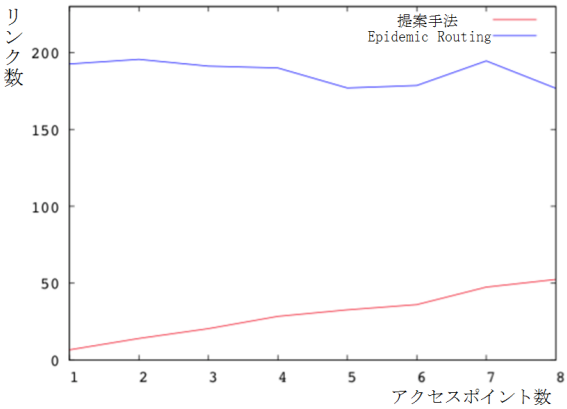

図 5　総リンク数の推移

Epidemic Routing の場合は総リンク数は約 180〜190 になり，端末の数の約 4 倍ものリンクが作られていることがわかる．それに対し提案手法では，アクセスポイントが 1 台の場合は約 7 であり，それからアクセスポイントを 1 台増やすごとにリンク数は 6〜8 ずつ増えていき，アクセスポイント数が 7 台以上

になると 50 を超えはじめた．つまり，端末の 1/10 以下の数のノードがやり取りをすることで全端末にリンクがつながったことがわかる．このことから，提案手法を用いてリンクの発生に条件をつけることで余分なリンクを抑えることができると確認できた．また，もっと端末の密度が高い場合は，より少ないアクセスポイントで全ての端末の通信を賄うことができると予想できる．しかし，リンク数を減らすことで通信機会が減少し，情報が伝搬しないといったことが起こってしまう可能性もある．よって，今後はこのようなことも考慮しつつ，既存研究や既存技術を本研究にいかすことができないか考えながら行っていく必要がある．

7. まとめと今後の課題

大規模災害によって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動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アプリを用いることでデータの同期ができそうだということが確認できた．また，提案手法と Epidemic Routing でノード間にリンクを発生させた場合を比較し，提案手法を用いることで余分なリンクを抑えらることができることを確認した．

今後の課題としては，通信ネットワークシミュレータを用いて本研究の評価を行い，リンク数と情報伝搬の関係を分析したり，より現実的な電波伝搬やモビリティを考慮していく．また，既存研究や既存技術を用いて本研究が拡張できないかを検討していきたい．

文　献

[1] NTT ドコモ
https://www.nttdocomo.co.jp/info/disaster/

[2] KDDI
http://www.au.kddi.com/mobile/anti-disaster/saigai-dengon/

[3] SoftBank
http://www.softbank.jp/mobile/service/dengon/

[4] Facebook
https://www.facebook.com/about/safetycheck/

[5] goo 防災アプリ
https://bousai.goo.ne.jp/apps/

[6] 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 (DICOM2013) シンポジウム，pp.964-969，2013 年 7 月.
http://www.fun.ac.jp/ y-nakamr/research/dicomo/dicomo2013kaneta.pdf

[7] 加藤寧 ”スマホ de リレー - 圏外でも通信可能なデュアルモードアドホックネットワーク技術 - ”，東北大学電気通信研究機構シンポジウム，2013 年 7 月.
http://www.roec.tohoku.ac.jp/info/news/pdf/NJ00067.pdf

[8] 鶴正人，内田真人，滝根哲哉，永田晃，松田崇弘，巳波弘佳，山村新也 ”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.

[9] IBR-DTN
https://bousai.goo.ne.jp/apps/