

# OpenFlowを用いたWebアプリケーションベースの 災害時救命情報収集機構の検討

多幡 早紀<sup>1,a)</sup> 上田 紘平<sup>1,b)</sup> 福井 良太郎<sup>2,c)</sup> 嶋津 恵子<sup>3,d)</sup> 重野 寛<sup>1,e)</sup>

概要：災害時に要救助者に関する情報の収集、蓄積を通じて救助活動を支援する、救命情報共有システムが検討されている。このシステムでは、公共車両に搭載される中継器を用いて情報収集を行う。中継器は、周囲の端末からの情報を gateway スイッチを経由してサーバへ送信する。中継器と gateway スイッチ間には、冗長化のため複数の通信回線が用意されている。しかし、それらの無線回線が全て切断した場合には、情報の損失が発生する。また、端末がネットワークに接続できないときには、情報を送信することができない。そこで本稿では、Web アプリケーションをベースとしたオフライン運用機構と、OpenFlow による経路制御機構からなる救命情報収集機構を提案する。提案機構では、キャッシングと同期処理により、オフライン時も端末からの情報送信を可能にする。また、中継器と gateway スイッチ間の通信を OpenFlow を用いて制御し、無線回線の接続状況や優先度に応じた回線選択を行う。全回線切断時には、中継器が受信した情報を車載の待機サーバに保持させることで、情報の損失を防ぐ。本稿では、OpenFlow による経路制御機構の実装を行い、サーバにおける情報収集率の向上という点から、提案機構の有用性を示す。

## 1. はじめに

大規模災害発生時には、被災地や被災者に関する情報の共有が不可欠である。しかし東日本大震災においては、重篤地域に関する情報の効率的な共有ができず、救命隊による被災者の救助開始が大幅に遅れた [1][2]。これは、現在の救助活動の体制では、要救助者に関する情報は口頭で伝達されるため、救命隊間で正確かつ迅速な情報共有が困難であるからである。

そこで現在、救命情報共有システム [3][4][5][6][7] が検討されている。このシステムの目的は、要救助者に関する情報の収集や蓄積を通して、救命隊による救助活動の早期開始を支援することである。このシステムにおける通信の特徴として、OpenFlow[8] による経路制御と、公共車両による情報の中継機能が挙げられる。

救命情報共有システムにおいて、救命隊や被災者から送

信される要救助者に関する情報は、全てサーバに蓄積される。このサーバが接続する有線ネットワークは、OpenFlow を用いて制御される。各地から無線回線を用いて送信された要救助者に関する情報は、gateway スイッチを経由し、この有線ネットワークに送信される。

公共車両に搭載される中継器は、スマートフォンなどの周囲の端末から送信された要救助者に関する情報を受信し、gateway スイッチを経由して、サーバへ送信する役割を担う。これは、公衆回線が利用できない地域においても、端末から情報を収集するための仕組みである。中継器は、無線 LAN や ARIB STD-T109[9] を利用し、基地局として周囲の端末から情報を受信する。災害時はネットワークが不安定であるため、サーバとの通信には回線の冗長化が求められる。そのため、中継器は gateway スイッチとの通信に、LTE や無線 LAN、衛星回線など、複数の無線回線が使用できるようになっている。このとき、複数存在する通信回線から、通信に使用する回線を適切に選択する必要がある。このように複数の通信回線が用意された場合でも、全ての回線が切断したときには、送信中の情報の損失が発生するという問題がある。また、中継器が端末の通信可能範囲に存在しないときには、端末から情報を送信することができない。

そこで本稿では、Web アプリケーションをベースとしたオフライン運用機構と、OpenFlow による経路制御機構が

<sup>1</sup> 慶應義塾大学大学院理工学研究所  
Graduate School of Science and Technology, Keio University  
<sup>2</sup> 慶應義塾大学先端研究センター  
Keio Advanced Research Center  
<sup>3</sup> 産業技術大学院大学  
Advanced Institute of Industrial Technology  
a) tabata@mos.ics.keio.ac.jp  
b) kohei@mos.ics.keio.ac.jp  
c) fukui@co-mobility.com  
d) shimazu-keiko@aait.ac.jp  
e) shigeno@mos.ics.keio.ac.jp

表 1 情報共有フレームワーク：METHANE

	名称	内容
M	Major of incident	大規模災害発令
E	Exact location	要救助者の存在場所
T	Type of incident	救出方法や医療処置の必要性
H	Hazard	現在/今後発生する危険
A	Access to scene	災害発生場所への到達経路
N	Number of casualties	要救助者の数
E	Emergency services	救出/救命に必要な装備

らなる、災害時の救命情報収集機構を提案する。提案機構では、キャッシングと同期処理により、オフライン時も端末からの情報送信を可能にする。また、中継器と gateway スイッチ間の通信を OpenFlow を用いて制御し、無線回線の接続状況や優先度に応じた回線選択を行う。全回線切断時には、中継器が受信した情報を車載の待機サーバに保持させることで、情報の損失を防ぐ。

以下本稿では、2章において救命情報共有システムの概要およびその通信における問題点を述べる。3章で提案機構について述べ、4章で OpenFlow による経路制御機構の動作確認の結果を示す。5章にて結論を述べる。

## 2. 救命情報共有システム

本章では、救命情報共有システムの概要を説明し、システムにおける通信の特徴および問題点について述べる。

### 2.1 救命情報共有システムの背景

災害時には「黄金の72時間」という概念がある。これは、災害発生から72時間が経過すると、要救助者の生存率が急激に減少するため、救助活動が72時間以内に行われる必要があることを指す。災害発生から24時間以内には、災害医療派遣チーム DMAT (Disaster Medical Assistance Team)[10] が被災地に到着する。しかし、その時点ではまだ救助活動に必要な情報が不足しているため、DMAT は即座に救助活動を開始することができない。現状では、DMAT は被災地に到着後、要救助者に関する情報を自力で収集し、その後救助活動を開始する。その結果、「黄金の72時間」の終盤によく救助活動のピークを迎える。そこで、要救助者に関する情報の収集を支援することで救助活動の早期開始および被災者の救命率を向上させることを目指す、救命情報共有システムが検討されている。

### 2.2 救命情報共有システムの概要

救命情報共有システムでは、NATO 軍により開発された METHANE[11] と呼ばれる情報共有フレームワークを用いて、要救助者に関する情報の収集・蓄積・提供を行う。METHANE により、最小限の情報量で救助活動に必要な情報を収集できる。METHANE の構造を表1に示す。

救命情報共有システムでは、現在口頭で伝達されている



図 1 救命隊による情報の閲覧

METHANE 情報の電子化を行い、サーバに蓄積することで、救命隊間の情報共有を支援する。救命隊は自身が所持する端末から、METHANE のフレームワークに沿った要救助者に関する情報や隊の活動地域の情報をサーバに送信する。また、被災者もサーバに情報を提供することが可能である。救命隊はサーバに蓄積された情報を閲覧することで、救助活動を必要とする被災地の状況を把握し、急行すべき地域を自律的に判断できる。図1に救命隊がサーバに蓄積された情報を閲覧する際の、端末の画面イメージを示す。地図の表示や情報の集約には、位置の特定が容易であり、全世界で使用されている MGRS グリッド [12] を採用した。地図上に、区画ごとに集計された METHANE 情報数を表示することで、重篤地域を直感的に把握できる。また、サーバに蓄積された情報を、救命隊だけではなく、地方自治体など災害対応にあたる関連機関にも公開することで、組織を超えた情報共有を実現できる。

### 2.3 Web アプリケーションの適用

救命情報共有システムの実装方法として、以下の2つの方法が考えられる。一つ目は、端末にインストールする専用アプリケーションとして実装する方法である。この方法では、端末内にシステムが保持されるため、ネットワークの状態に影響されることなく、システムを操作することが可能である。しかし、システムの動作が端末の環境に依存するため、様々な種類の端末で動作させる場合には適していない。また、システムの利用を開始する前に、端末にシステムをインストールする必要がある。二つ目は、Web アプリケーションとして実装する方法である。この方法では、

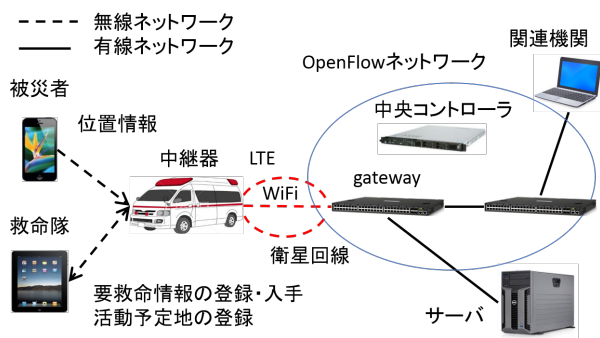


図 2 救命情報共有システムの通信環境

端末にシステムを事前にインストールする必要がなく、即座にシステムを利用できるという利点がある。また、端末の種類や環境によらず動作させることができる。Webアプリケーションを操作するには、基本的に端末が通信可能でなければならないが、近年では、アプリケーションキャッチ [13] など、オフライン時も Web アプリケーションを動作させることができる仕組みが普及している。以上の理由から、救命情報共有システムは Web アプリケーションとして提供される。

## 2.4 救命情報共有システムの通信

救命情報共有システムにおいて、救命隊や被災者の端末が、情報を蓄積するサーバと通信を行う様子を図 2 に示す。要救助者に関する情報が蓄積されるサーバは、OpenFlow により制御される有線ネットワークに接続されることを想定する。OpenFlow とは、SDN (Software Defined Network) を実現する技術の一種であり、従来の IP ネットワークと比較し、柔軟な経路制御が可能である。本システムにおいて OpenFlow を導入することで、情報の重要度やネットワークの状況に応じた経路制御が実現できる。例えば、サーバ宛てのパケットを複製し、それぞれ異なる経路を使用して送信することで、パケットのサーバへの到達率を向上させることができる。また、帯域に余裕のある経路を優先的に使用することも可能である。各地から無線回線を使用して送信された情報は、gateway スイッチを経由し、有線ネットワークを使用してサーバに送信される。

また、救命情報共有システムを実現するうえで、公共車両に中継器を搭載することが考えられる [3]。これは公共車両が被災地を移動することを利用して、端末から公衆回線を用いて通信ができない地域において、情報を収集するための仕組みである。700MHz 帯を用いた車車間・路車間通信の標準規格として、ARIB STD-T109[9] があり、今後の普及が期待できる。そこで、公共車両に搭載された T109 無線機を、災害時にモードを切り替えて移動基地局として動作させることで、周囲の端末や車両と通信を行うことができる。もちろん、同様のシステムを無線 LAN を用いて構築することも可能である。中継器は T109 や無線 LAN

により受信した情報を、gateway スイッチへ送信する。このとき、中継器は gateway スイッチとの通信に、LTE や無線 LAN、衛星回線など、複数の無線回線が使用可能である。これは災害時にはネットワークが不安定であることを考慮し、中継器と gateway スイッチ間の通信の可能性を高めるためである。

## 2.5 救命情報共有システムの通信の問題点

救命情報共有システムを運用するうえで、災害時の不安定な通信環境においても、より多くの要救助者に関する情報をサーバに収集し、救命隊に提供することが重要である。本システムを不安定な通信環境において利用する場合、次の 2 点を考慮しなければならない。

第一に、本システムは Web アプリケーションであるため、端末がネットワークに接続していなければ、救命隊は情報の送信や閲覧を行うことができない。中継器が端末の通信可能範囲に存在しない場合に、端末に情報を入力することができず、救命隊による情報提供の機会を妨げることになり、サーバに情報が蓄積されない。そのため、端末の通信状態によらずに、救命隊が情報提供できることが理想である。そこで、端末がオフライン状態である場合において、本システムを部分的にでも操作できるようにする必要がある。

第二に、中継器と gateway スイッチ間において、通信回線の状況に応じて、使用する回線を動的に変更しなければならない。中継器と gateway スイッチ間は、冗長化のため複数の無線回線で接続されている。そのため、ある回線が切断しても、他の通信可能な回線を用いて情報を送信することが考えられる。この機能を実現するためには、無線回線の状況に応じて、通信に使用する回線を動的に切り換える機構が必要である。また、中継器がサーバと通信中であるときに回線が切断した場合、送信中のパケットは破棄され、サーバまで到達しない。情報の損失を防ぐためには、中継車両において、通信が完了するまで送信する情報を保持する機構が必要である。

## 3. 提案

本章では、救命情報共有システムにおける、Web アプリケーションをベースとしたアプリケーションレベルのオフライン運用機構と、OpenFlow によるネットワークレベルの経路制御機構からなる、情報収集機構を提案する。

### 3.1 提案の概要

提案機構の目的は、救命情報共有システムにおいて、端末とサーバ間の安定した通信を実現し、サーバにおける情報の収集率を向上させることである。そこで提案機構では、有線ネットワークを制御する OpenFlow コントローラ（以下、中央コントローラ）の制御範囲を中継器まで拡大し、

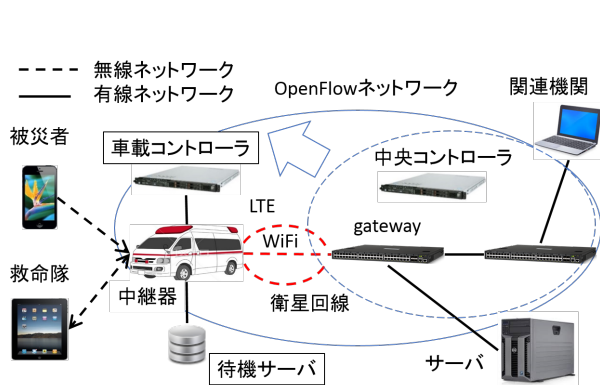


図 3 提案機構における通信環境

中継器と gateway スイッチ間の通信に使用する無線回線の動的な切り換えを可能にする。また、中継器と gateway スイッチを接続する全ての回線が切断することを想定し、中継器の制御を行う車載コントローラと、中継器が受信した情報を保持する待機サーバを中継器に接続する。提案機構は次の 2 つの機構から構成される。

一つ目は、救命情報共有システムのオフライン運用機構である。アプリケーションキャッシュや WebSQL [14] を用いて、ネットワークの切断中に端末に入力された METHANE 情報を保持できるようにする。端末内に蓄積された情報は同期処理によってサーバに送信される。

二つ目は、中継器と gateway スイッチ間の無線回線の切断に備えた経路制御機構である。中央コントローラと車載コントローラは、中継器と gateway スイッチ間の各無線回線を監視し、回線の状態の変化を検知する。そして、回線の切断や復旧、優先度の変化に応じて、中継器と gateway スイッチのフローテーブルを更新する。また、中継器と gateway スイッチ間の全ての回線が切断したとき、車載コントローラにより、中継器が周囲の端末から受信したパケットを待機サーバに転送することで、中継車両による情報の保持を実現する。

### 3.2 ネットワーク構成

提案機構におけるネットワークの構成を図 3 に示す。救命情報共有システムの通信において OpenFlow による経路制御が行われるのは、gateway スイッチとサーバを接続する有線ネットワーク内のみである。しかし提案機構では、OpenFlow の適用範囲を中継器と gateway スイッチを接続する無線通信区間まで拡張する。その結果、中継器が中央コントローラの管理下に入り、有線ネットワーク内に加え、中継器と gateway スイッチ間の通信も、OpenFlow により動的に制御することが可能になる。

また、中継器と gateway スイッチ間の全ての回線が切断することを想定し、車載コントローラと待機サーバを導入する。車載コントローラは、中継器と gateway スイッチ間の全ての回線が切断したとき、中継器との通信ができなくなる中央コントローラの代わりに、中継器の制御を行う。

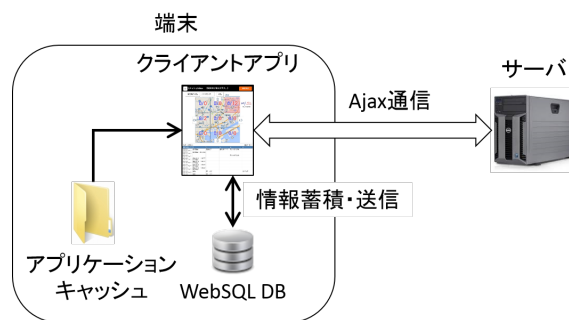


図 4 オフライン運用機構

車載コントローラを中継器に直接接続しておくことで、常に中継器の制御を行うコントローラが存在する状況を実現することができる。一方待機サーバは、中継器が gateway スイッチと通信できない期間に、中継器が周囲の端末から受信した情報を保持するためのものである。車載コントローラと待機サーバを中継器に接続することで、中継器が孤立している間に要救助者に関する情報が損失することを防ぐことができる。

### 3.3 オフライン運用機構

提案機構では、通信回線の切断による情報損失を防ぐため、端末のブラウザ内に情報を保持する機構を導入する。図 4 にオフライン運用機構の動作を示す。

#### 3.3.1 アプリケーションキャッシュと WebSQL の利用

端末がネットワークに接続できない状況においても本システムを操作できるようにするため、アプリケーションキャッシュを利用する。アプリケーションキャッシュとは、HTML5 に導入されている、システムの一部をブラウザキャッシュに保持する仕組みのことである。この仕組みでは、端末がはじめてサーバに接続したときにブラウザのローカル環境にキャッシュさせるファイルを、マニフェストファイル内であらかじめ定義しておく。提案機構では、マニフェストファイル内で、救命情報共有システムを全てローカル環境にて保持することを定義する。その結果、端末は一度サーバと接続し、必要なファイルをキャッシュした後は、本システムをネットワークの状況に関わらず操作することができる。

また、提案機構では、WebSQL を用いてサーバと同様のデータベースをブラウザ内に構築する。ブラウザにデータベースを構築することで、端末がネットワークに接続していない期間に入力された情報を、ブラウザ内に保持しておくことが可能である。端末に入力された情報は、直接サーバに送信するのではなく、一度 WebSQL データベースに挿入してから、次に示す同期処理を用いて送信する。

#### 3.3.2 同期処理

同期処理では、WebSQL データベースに蓄積された METHANE 情報を、Ajax (Asynchronous JavaScript + XML) 通信 [15] を用いてサーバへ送信する。Ajax 通信は、ブラウザとサーバ間でデータ連携を行うための非同期通信

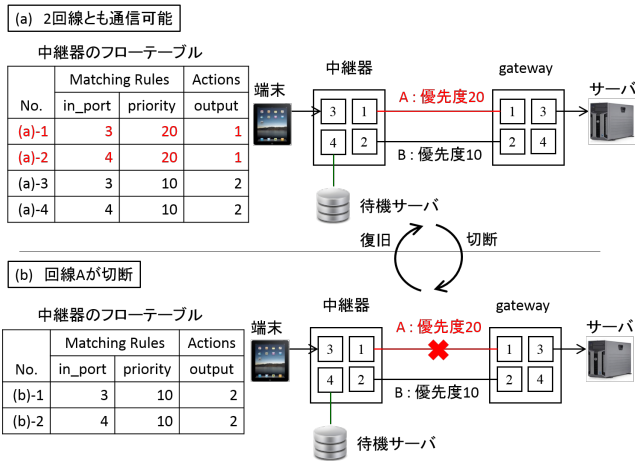


図5 回線の接続状況の変化に応じた回線選択

方式である。提案機構では、端末がネットワークに接続可能であると検知した場合に、Ajax 通信を用いて METHANE 情報の送信を行う。サーバは端末からの情報を受信し、データベースに蓄積した後、端末に更新完了メッセージを送信する。端末はサーバからの更新完了メッセージを受信すると、WebSQL データベースから、サーバに蓄積された情報を削除する。ここで、端末は情報の送信後、更新完了メッセージを一定時間以内に受信しない場合には、情報の再送信を行う。つまり、端末はサーバに情報が確実に蓄積されたことを確認するまで、WebSQL データベース内に情報を保持する。この処理により、ネットワークが切断中でも情報を損失することなく、ネットワークが復旧した後、サーバに蓄積することができる。

### 3.4 経路制御機構

図3に示した通信環境において、提案機構では OpenFlow の仕組みを用いて、中継器と gateway スイッチ間の無線回線の状況に応じて、通信に使用する回線の切り換えを行う。また全回線切断時には、待機サーバにおいて周囲の端末から受信した情報を保持し、情報の損失を防ぐ。

#### 3.4.1 回線の接続状況の変化に応じた回線選択

提案機構では、中央コントローラ、車載コントローラとともに、中継器と gateway スイッチ間の各通信回線の監視を行う。そして、回線の接続状況が変化するとき、中継器と gateway スイッチのフローテーブルを更新する。図5に回線の接続状況が変化した場合の回線選択の例を示す。図5(a)では、中継器と gateway スイッチを接続する回線 A, B がどちらも通信可能な状態である。このとき回線 A が切断すると、(b)に示すように、フローテーブルから回線 A に関するフローエントリ (a)-1, (a)-2 が削除される。

各コントローラは独立して、中継器と gateway スイッチ間の各通信回線の接続状況の監視を行う。回線の切断の判断にはタイムアウト方式を用いる。図6に提案機構における、中継器と gateway スイッチ間のリンクの検出例を示す。

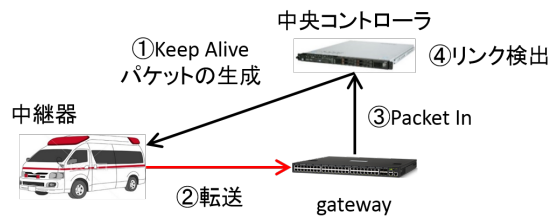


図6 Keep Alive パケットを用いたリンクの検出例

各コントローラは、あらかじめ中継器と gateway スイッチのフローテーブルに対し、Keep Alive パケットの受信時に Packet In を発生させるフローエントリを追加する。ここで Packet In とは、OpenFlow ネットワークにおいて、スイッチが受信したパケットをコントローラに送信し、処理を問い合わせることを指す。Packet In メッセージを解析することで、パケットを受信したスイッチとそのポートを特定することができる。各コントローラは、中継器と gateway スイッチ間を接続する各無線回線に対し、定期的に Keep Alive パケットを生成して送信することで、回線が通信可能であることを確認する。中央コントローラは、回線が切断したと判断したとき、該当する回線に関わるフローエントリを削除する。また、回線の復旧を検知した場合には、復旧した回線を使用するフローエントリを追加する。この動作により、中継器と gateway スイッチは、通信可能な回線の一覧を常に最新の状態で所持することができる。

#### 3.4.2 回線の優先度の変化に応じた回線選択

中央コントローラは、定期的中継器と gateway スイッチ間の各通信回線の優先度を更新する。そして、各回線の優先度に変化が生じた場合、中継器と gateway スイッチのフローテーブルを更新する。なお中央コントローラは、回線の監視や外部からの情報により、各回線の優先度を決定できるものとする。以下、回線の優先度が変化した際の回線選択の動作を説明する。

中央コントローラは、中継器と gateway スイッチ間の各通信回線の優先度の一覧を所持している。そして、定期的に新たな優先度の一覧を作成し、古い優先度の一覧との比較を行う。優先度に変更があった場合、中央コントローラは、中継器と gateway スイッチのフローテーブルから、古い優先度の値をもつフローエントリを削除し、新たな優先度の値をもつフローテーブルを追加する。この動作により、各回線の状況の変化を迅速にパケットの処理に反映させることができる。

#### 3.4.3 中継車両における情報の保持

端末と中継器が通信可能である場合でも、中継器と gateway スイッチが通信可能でなければ、中継された情報は損失する。そこで、中継器と gateway スイッチを接続する全ての無線回線が切断したとき、中継器が周囲の端末から受信する要救助者に関する情報を損失させないために、中継器に接続された待機サーバに情報を保持させる。このと

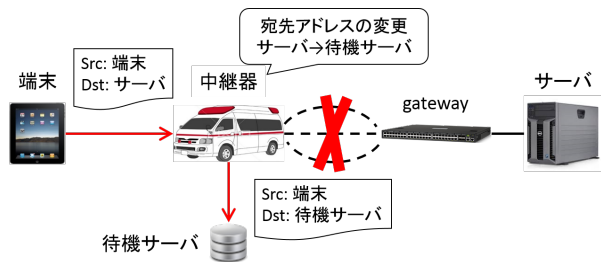


図 7 中継車両における情報の保持

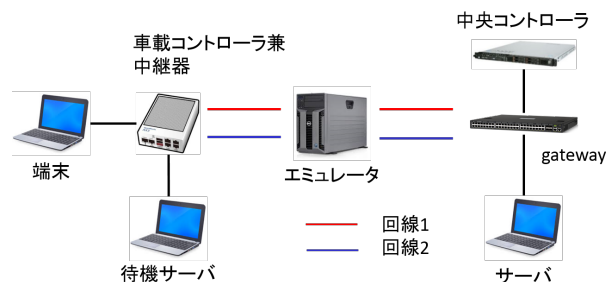


図 8 実験環境

き、中央コントローラは中継器と通信することができないため、車載コントローラが中継器の制御を行う。図 7 に、中継車両において、サーバ宛てのパケットを待機サーバに誘導し、蓄積させる仕組みを示す。以下、中継車両における情報の保持機構について、中央コントローラと車載コントローラの動作に分けて説明する。

中央コントローラの動作は、回線の接続状況が変化した際の回線選択時と同様である。中央コントローラは、回線の切断に応じて、gateway スイッチのフローテーブルから該当する回線に関わるフローエントリを削除する。

一方車載コントローラも、中継器と gateway スイッチ間の全ての回線の切断を確認した場合、中継器のフローテーブルから該当する回線に関わるフローエントリを削除する。そして、中継器が周囲の端末から受信したパケットを待機サーバに送信するフローエントリを追加する。このとき、受信したパケットの宛先をサーバから待機サーバへと変更するため、出力ポートの指定だけではなく、宛先の IP アドレスと MAC アドレスを待機サーバへと書き換える処理も同時に行う。中継器が宛先のアドレスを書き換えることで、端末が意図することなく、待機サーバにパケットを誘導することができる。この動作により、中継器と gateway スイッチが通信できない期間に、要救助者に関する情報の損失を防ぐことが可能である。

待機サーバに蓄積された情報は、中継器と gateway スイッチ間の通信が復旧した際に、再びサーバに送信される必要がある。このとき、待機サーバからサーバに情報を送信する方法として、以下の 2 つの方法の検討を進めている。

一つ目は、中継器と gateway スイッチ間の回線状況にかかわらず、定期的に情報を送信するケースである。このケースは、車載コントローラに機能を追加する必要がなく、待機サーバが情報の送信を繰り返すことだけで実現できる。しかし、中継器と gateway スイッチが通信できない状態が継続した場合には、無駄な送信が幾度も発生する。

二つ目は、待機サーバが車載コントローラの指示を受けて、情報を送信するケースである。このケースでは、中継器と gateway スイッチが通信可能であることを確認したうえで情報を送信することができるため、無駄な送信が発生しない。しかし、車載コントローラと待機サーバが連携するためには、待機サーバによる情報の送信を許可するメッ

表 2 実験環境の構成

車載コントローラ兼中継器	OpenBlocks AX3 (OS: Debian 7.7)
gateway スイッチ	Pica8 P-3297
中央コントローラ	IBM x3250 M4 (OS: Ubuntu 13.04)
SDN フレームワーク	Ryu ver. 3.15
ソフトウェアスイッチ	Open vSwitch ver. 2.0.2

セージを定義しなければならない。また、そのメッセージの取り扱いについて、車載コントローラ、待機サーバともに機能の追加が必要である。

#### 4. プロトタイプ実装と動作確認

本章では、OpenFlow による経路制御機構の実装とその動作確認について述べる。

##### 4.1 実験環境

提案機構の実装には Ryu[16] を用いた。回線の接続状況の検知に必要な Keep Alive 機能として、中継器と gateway スイッチ間で、定期的に ICMP メッセージの送受信を行うように実装した。また、今回は優先度の指標として回線のパケットロス率を採用した。前回の優先度の更新から今回の更新までの間の ICMP メッセージの受信数をもとに、優先度の算出を行った。

実験環境を図 8 に、使用した機材を表 2 に示す。中継器は Open vSwitch[17] を用いて実装し、さらに Ryu を導入することで、車載コントローラとしても動作させた。本稿では、中継器と gateway スイッチ間が 2 回線で接続されていることを想定し、それぞれ異なるパケットロス率を通信エミュレータを用いて設定した。パラメータは表 3 に示す値を用いた。この実験環境において、端末からサーバに送信されるパケットを Iperf[18] により生成し、以下の項目の確認および測定を行った。

- エミュレータにおける各通信回線の使用状況
- サーバおよび待機サーバにおけるパケットの到達数

実験では、15 秒間隔で各通信回線の接続状況をエミュレータにより変化させた。回線の切断は、パケットロス率を 100% にすることで実現した。測定期間は各回線の状態

表 3 実験パラメータ

トラフィックの種類	UDP
送信速度	1Mbps
実験時間	75 sec
回線切断判定までの <i>timeout</i>	1 sec
ICMP メッセージの送信間隔	1 sec
優先度の更新間隔	10 sec
回線 1 のパケットロス率	1 %
回線 2 のパケットロス率	10 %

表 4 実験シナリオ

区間	A	B	C	D	E
経過時間 [sec]	0-15	15-30	30-45	45-60	60-75
回線 1 の状態	○	×	×	×	○
回線 2 の状態	○	○	×	○	○

により、5つの区間に分類することができる。各区間における回線 1, 2 の状態を表 4 に示す。接続状態を○、切断状態を×で表現している。以下、表中の A~E を利用して測定結果を説明する。

#### 4.2 回線の使用状況

実験において、エミュレータの各回線を通過したパケットをプロットしたものが図 9 である。回線 1, 2 がともに接続状態にある区間 A では、回線 1 が使用されている。これは、パケットロス率が低い回線 1 の方が優先度が高いと中央コントローラが判断しているからである。回線 1 が切断して区間 B になると、使用される回線が回線 2 に切り換わっている。区間 C ではどちらの回線も切断されているため、中継器と gateway スイッチは通信することができない。区間 D になり回線 2 が復旧すると、再び回線 2 を使用してパケットが送信されている。区間 E においては、使用する回線が途中で回線 2 から回線 1 に切り換わっている。これは、回線 1 の復旧後初めてとなる優先度の更新が行われた際に、回線 1 と回線 2 の優先度が逆転したからである。以上より、回線の接続状況や優先度の変化に応じて、使用する回線が切り換わることが確認できた。

図 9 において、区間 A から区間 B への移行および区間 B から区間 C への移行の際、回線の接続状況が変化してから、使用する回線が切り換わるまでに時間差があることが読み取れる。これは、提案機構では回線の接続状況の検知に少なくとも *timeout* 時間を要するためである。

#### 4.3 サーバと待機サーバにおけるパケットの到達数

実験において、サーバおよび待機サーバにおけるパケットの受信数を示したものが図 10 である。回線 1, 2 のいずれかが接続状態にある区間 A, B, D, E では、サーバにおけるパケットの受信数が増加していく。待機サーバは中継器と gateway スイッチ間が通信可能であるときには必要がないため、待機サーバにおけるパケットの受信数は変化し

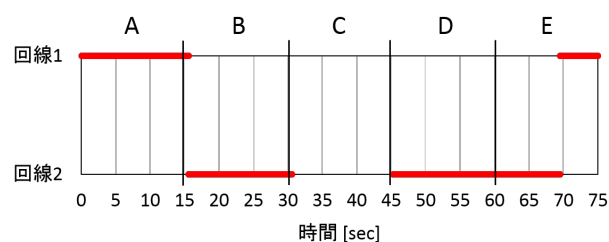


図 9 エミュレータにおける各回線の使用状況

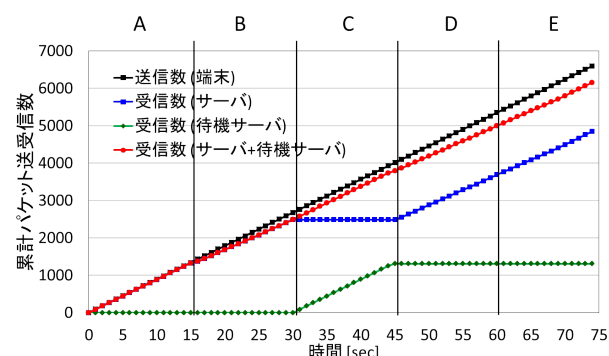


図 10 サーバおよび待機サーバのパケットの受信数

ない。一方、回線 1 と回線 2 がともに切断している区間 C では、サーバにおけるパケットの受信数が変化しないのに対し、待機サーバでは受信数が増加している。このことから、中継器と gateway スイッチが通信できない間、中継器が受信したパケットはサーバではなく待機サーバに送信されていることが読み取れる。従って、中継器両における情報の保持機構により、パケットの損失を防ぐことが可能であることが確認できた。

また、サーバと待機サーバにおけるパケットの受信数を加えた値に着目すると、端末による送信数と比較して、ほぼ同等の値を示している。待機サーバに保持された情報が、中継器と gateway スイッチ間の通信が復旧したときに改めてサーバに送信されることを考慮すると、提案機構によりサーバにおいて高い情報収集率を実現できることが確認できた。なお、端末による送信数とサーバと待機サーバの合計受信数が完全に一致しないのは、フローの切り換えの際のパケットロスに加え、回線 2 のパケットロス率が大きいためであると考えられる。

ここまで通信トラフィックとして UDP パケットを用いた結果を示したが、実験では TCP パケットを用いた場合も、提案機構によりフローの切り換えが行われていることを確認した。しかし、セッション中に中継器と gateway スイッチ間の通信が切断された場合、セッションを継続することができず、情報の損失が発生する。対策については、今後検討する予定である。

## 5. おわりに

本稿では、救命情報共有システムにおける、Web アプリケーションをベースとしたアプリケーションレベルのオフライン運用機構と、OpenFlow によるネットワークレベルの経路制御機構からなる、情報収集機構を提案した。提案機構では、アプリケーションキャッシュや WebSQL によるキャッシングと、Ajax 通信による同期処理により、オフライン時も端末からの情報送信を可能にした。また、中継器と gateway スイッチ間の通信を OpenFlow を用いて制御し、無線回線の接続状況や優先度に応じた回線選択を行った。中継器と gateway スイッチ間の全ての無線回線が切断したときには、中継器の制御を行う車載コントローラにより、中継器が周囲の端末から受信した情報を待機サーバに保持させることで、情報の損失を防いだ。

実験環境を構築し、提案機構の OpenFlow による経路制御機構が正しく動作することを確認した。中継器が gateway スイッチと通信できない場合に、中継器が受信したパケットを待機サーバに送信することで、情報の損失率を著しく低下させた。以上より、中継器と gateway スイッチ間の安定した通信と、サーバにおける情報の収集率の向上という点で、提案機構の有用性を確認した。

謝辞 本研究の一部は、総務省戦略的情報通信研究開発推進事業 (SCOPE) 先進的通信アプリケーション開発推進型研究開発の支援により行われました。

## 参考文献

- [1] 小滝 晃：東日本大震災緊急災害対策本部の 90 日 - 政府の初動・応急対応はいかになされたか -, ぎょうせい (2013).
- [2] Kitsuta, Y., Niiyama, S., Ushijima, K., Nakajima, S., Gunshin, M., Ishii, T., Nakamura, K., Matsubara, T., Yamaguchi, D., Katada, S. and Komatsu, K.: Usefulness of modified METHANE report as the communication method between hospital and medical team during the East Japan Earthquake, *Japanese Journal of Trauma and Emergency Medicine*, Vol. 3, No. 1, pp. 5-12 (2012).
- [3] 福井良太郎, 嶋津恵子, 重野 寛：大規模災害急性期サーチ・アンド・レスキュー支援システム, 情報処理学会研究報告 ITS, Vol. 2014, No. 3, pp. 1-6 (2014).
- [4] 嶋津恵子：準天頂衛星を通信基盤とした災害急性期救命支援システムのデザイン構想 - NATO 開発 METHANE レポートの国際対応版への拡張-, 電子情報通信学会技術研究報告, Vol. 114, No. 87, pp. 85-90 (2014).
- [5] 多幡早紀, 堂ノ脇梓, 福井良太郎, 嶋津恵子, 重野 寛：OpenFlow を用いた災害時の動的な回線選択手法の検討, 情報処理学会第 77 回全国大会, pp. 3.51-3.52 (2015).
- [6] 堂ノ脇梓, 多幡早紀, 嶋津恵子, 福井良太郎, 重野 寛：不安定なネットワークを想定した救命情報共有システムのためのオフライン運用機構, 情報処理学会第 77 回全国大会, pp. 3.49-3.50 (2015).
- [7] 多幡早紀, 上田紘平, 福井良太郎, 嶋津恵子, 重野 寛：車載中継器のための OpenFlow を用いた災害時の動的な回線選択機構, 情報処理学会マルチメディア, 分散, 協調と

- モバイル (DICOMO 2015) シンポジウム, pp. 1525-1532 (2015).
- [8] Open Network Foundation: OpenFlow - Open Network Foundation, <https://www.opennetworking.org/sdn-resources/openflow> (2015).
- [9] 電波産業会：700MHz 帯高度道路交通システム標準規格 / ARIB STD-T109 1.0 版 (2012).
- [10] DMAT 事務局：DMAT 事務局ホームページ, <http://www.dmat.jp/> (2015).
- [11] Col. Dr. Ingo Hartenstein: Medical Evacuation in Afghanistan: Lessons Identified! Lessons Learned?, *Medical Challenges in the Evacuation Chain* (2008).
- [12] National Geospatial-Intelligence Agency: NGA: DMA TECHNICAL MANUAL 8358.1, <http://earth-info.nga.mil/GandG/publications/tm8358.1/tr83581b.html> (2015).
- [13] W3C: Offline Web applications, <http://www.w3.org/TR/2011/WD-html5-20110525/offline.html> (2011).
- [14] W3C: Web SQL Database, <http://www.w3.org/TR/webdatabase/> (2010).
- [15] Garrett, J. J.: Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/> (2005).
- [16] Ryu SDN Framework Community: Ryu SDN Framework, <http://osrg.github.io/ryu/> (2015).
- [17] Open vSwitch: Open vSwitch, <http://openvswitch.org/> (2015).
- [18] Iperf. fr: Iperf - The TCP / UDP Bandwidth Measurement Tool, <https://iperf.fr/> (2015).