

プローブデータ解析のためのアーキテクチャ提案

2010SE002 秋元 累 2010SE053 五十川 雄太

指導教員：青山 幹雄

1 はじめに

近年、車は私たちの生活に欠かすことができないものになっている。しかし交通渋滞が日常的な現象になり、ドライバが運転中に直面する問題となっている。

本研究では、位置情報などのプローブデータを解析し、広範囲で、渋滞を緩和するアーキテクチャを提案する。

2 研究の背景

高度道路交通システム(ITS: Intelligent Transport System)の構築が世界的に行われている。近年では、プローブ情報システムという車両をセンサとして活用し、大量の情報を収集するシステムに注目が集まっている。

しかし、これらのシステムには、接続率の低さからくるノードの少なさ、情報の増大化による処理の複雑さ、費用問題から普及率の低さが課題となっている。そのため上記問題を解決するためアーキテクチャが必要となる。

3 研究の課題

本研究では上記の背景を踏まえ、以下 2 点を研究の目的とし、交通情報を考慮したアーキテクチャを提案する。

- (1)交通情報システム(VICS: Vehicle Information and Communication System)でサポートされていない道路でも、交通情報、天気情報を提供する。
- (2)MANET と複合イベント処理との連携に必要なプロトコルを定義する。

VICS は、情報の提供がセンサインフラの存在する区間に限られる課題がある。また車車間ネットワークと複合イベント処理との連携プロトコルが決定されておらず連携できない課題がある。

4 関連研究

4.1 PD(Probe Data)

PD とは、走行中の車両がそれぞれセンサとなり時刻、現在地などの情報を送信するものである[5]。PD 活用例として、インターナビプレミアムがある[2]。

4.2 MANET(Mobile Ad Hoc Network)

MANET とは、専用の基地局やアクセスポイントに依存せずに、モバイル端末自体が持つ中継機能を利用することによって一時的に相互接続される端末群で構成されるネットワークのことである[4]。

4.3 CEP(Complex Event Processing)

CEP は、リアルタイムにセンサから発生する大量のデータをリアルタイムに分析、判断、処理する技術である[1]。予め、判断するシナリオをメモリ上に展開しておき、イベントストリームに対してリアルタイムにシナリオを当てはめ分析していく[3]。

5 アプローチ

前述の課題を解決するために、MANET と CEP を連携させ、PD をリアルタイムに表示、処理する RISS(Road Info Supply System)アーキテクチャを提案する。MANET、CEP の連携に特に重みをおき、連携プロトコルの定義を行う。

これにより、センサインフラの存在を考慮することなく身近な車とは車車間通信を形成し、交通情報をリアルタイムに交換する。また MANET 外の車両も情報取得を可能とするため、交通情報をクラウド上にリアルタイムで配信し、格納するために CEP 技術を用いる。これにより、リアルタイムな交通情報取得、配信を実現する。

6 提案アーキテクチャ

6.1 RISS アーキテクチャ

RISS アーキテクチャを以下に示す(図 1)。

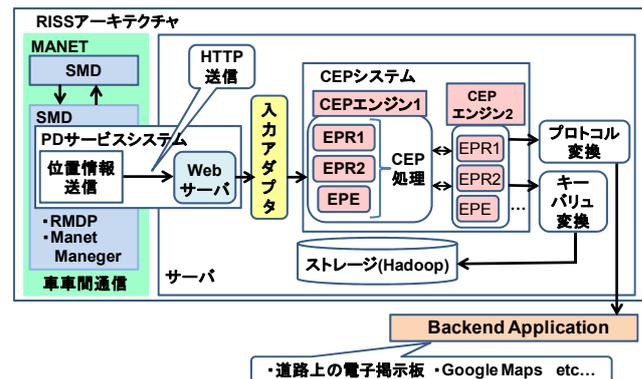


図 1 RISS アーキテクチャ

提案アーキテクチャの構成要素は、MANET 通信と CEP から形成される。MANET を車車間で形成し、形成されたネットワークで情報を交換する。以下に処理を説明する。

- (1)MANET ソフトウェアで形成された車車間ネットワークで、リアルタイムの道路情報、天気情報を安価に、広範囲に交換する。
- (2)MANET ソフトウェアを用いて、LTE 回線を介して車車間ネットワークで作成された道路、天気情報を CEP システムに送信する。その際、MANET で作成されたデータはプロトコル変換される。変換されたデータは、CEP システムに入る。
- (3)CEP システムに入ったデータは、CEP エンジン 1, CEP エンジン 2, ..., CEP エンジン n を通り CEP 処理される。
- (4)CEP システムで処理されたデータは、再びプロトコル変換し Googlemaps などの外部アプリケーションに送信され、キーバリュ変換し Hadoop に送信され、データ

マイニングが行われる。

6.2 提案アーキテクチャの構成要素

6.2.1 MANET

Manetmanager と PD サービスシステムから成り、Manetmanager は MANET を形成し、PD サービスシステムは取得した PD を Web サーバに POST 送信する。

6.2.2 入力アダプタ

MANET から送信されてきた PD を CEP システムで処理できる形に変換する。OSI 参照モデルに基づくプロトコル定義を行い、入力アダプタとする。

6.2.3 CEP システム

受信されたらすぐに処理されるべきイベントを、最小遅延で処理実行することを中心としたソフトウェアアーキテクチャである。CEP エンジン、EPR(Event Processing Rules)、EPE(Event Processing Engine)の 3 つで構成される。以下それぞれ説明していく。

(1) CEP エンジン

ストリーム処理を行うエンジンである。これらは EPR によって構成される。EPE も含まれる。CEP エンジンは、二つの異なった EPR セットで構成される。

(2) EPR

統計学などからの専門的知識や、以前検知され分析された結果から定義されたシナリオのことである。またフィルタリング、集約、階層化された上位のイベントの発生を満たす度に実行されるアクションも含まれる。

(3) EPE

CEP エンジン内の異なった EPR を実行することと、それに関連したアクションを実行する。

6.2.4 キーバリュ変換

CEP システムで処理されたデータを Hadoop で処理可能にするため、キーバリュ形式に変換して出力する。

6.2.5 プロトコル変換

CEP システムで処理されたデータを、Google Maps、道路上の電光掲示板といった外部アプリケーションで処理可能にするためにプロトコル変換を行う。

6.3 アーキテクチャの振る舞い

提案する RISS アーキテクチャの振る舞いを図 2 のシーケンス図に示す。

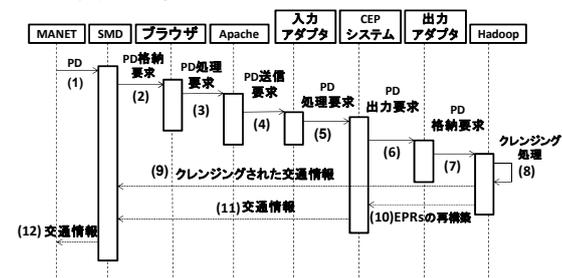


図 2 RISS アーキテクチャの振る舞い

- (1) MANET 環境で作られた PD を、SMD に送信する。
- (2) SMD は、MANET 環境から受信した PD をブラウザに格納する。
- (3) ブラウザは、格納された PD を Web サーバである Apache に送信する。
- (4) サーバに格納されている PD を、CEP システムの入力アダプタに送信し、PD 送信要求する。
- (5) 入力アダプタに送信されてきた PD を、CEP システム

に対し、PD 処理要求する。

- (6) CEP システムで処理された PD を出力アダプタに出力する、PD 出力要求をする。
- (7) 出力アダプタに出力された PD は、Hadoop にキーバリュ型に変換され格納される、PD 格納要求する。
- (8) 格納要求された PD はクレンジング処理され、Hadoop に格納される。
- (9) Hadoop のストレージでクレンジングされた PD は、交通情報として SMD に返される。
- (10) Hadoop のストレージに格納される PD は、クレンジング処理により、新たな EPRs として定義され、CEP システム内の CEP エンジンに返される。
- (11) CEP システムで処理された PD は、リアルタイムに外部アプリケーションなどに送信され、SMD に返される。
- (12) SMD に返された交通情報は、MANET 環境に返される。

7 提案アーキテクチャのプロトタイプ

MANET と、CEP システムを用いる PD 解析アーキテクチャに基づき実装する。

7.1 利用シナリオ

PD サービスシステム、入力アダプタのプロトコル定義の 2 つをプロトタイプとする。プロトタイプの利用シナリオとして、位置情報といった PD を Web サーバに送り、Web サーバで PD を受信する「PD サービスシステム」と、MANET と CEP システムを結合する入力アダプタのプロトコル定義の 2 つについて考えた。

7.1.1 PD 送信サービスのシナリオ

スマートモバイルデバイス上で取得された位置情報を Web サーバに送信するサービスである。PD を Web サーバに送ることで、CEP システムでリアルタイム処理を行う。MANET 環境から CEP システムへ PD を送信する際、HTTP 通信の POST メソッドを用いて PD を、Web サーバに送信する。

このシステムのユースケース(図 3)、状態遷移図(図 4)、シーケンス図(図 5)をそれぞれ以下に示す。

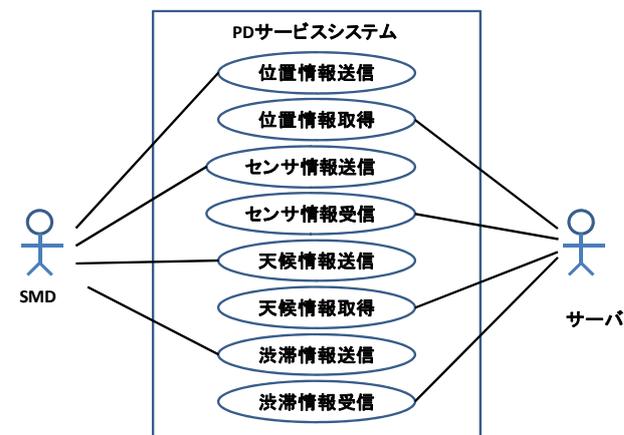


図 3 プロトタイプでのユースケース図

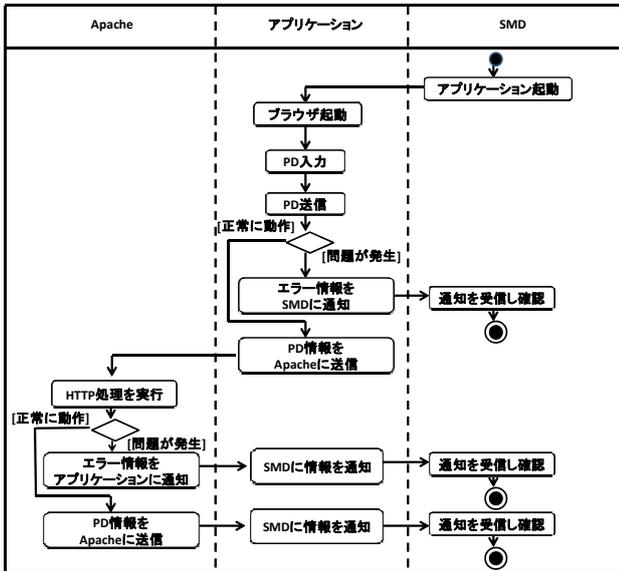


図4 プロトタイプの状態遷移図

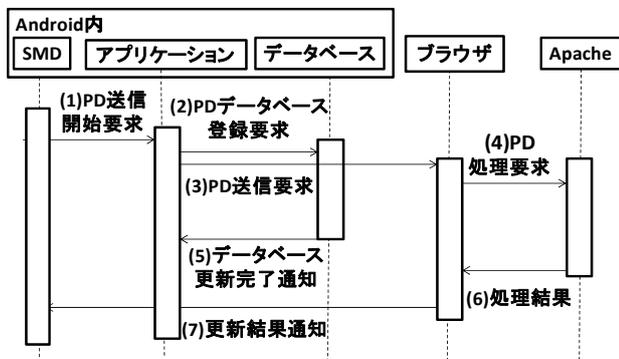


図5 プロトタイプのシーケンス図

7.1.2 入力アダプタのプロトコル定義

入力アダプタのプロトコル定義は、MANET 環境と、CEP システムを組み合わせるためのプロトコルを各レイヤで定義した。以下に選定要素を示す(表 1)。

表 1 プロトコル定義

レイヤ	選択したプロトコル
(1)アプリケーション層	HTTP
(2)プレゼンテーション層	HTTP
(3)セッション層	HTTP(Cookie)
(4)トランスポート層	UDP
(5)ネットワーク層	IP
(6)データリンク層	MAC, RLC, PDCP
(7)物理層	PHY

(1) アプリケーション層:HTTP

HTTP は、Web ブラウザと Web サーバの間で HTML などのコンテンツを送受信するプロトコルである。Socket といった通信プロトコルの実装と比べて、簡単に実装できる。POST メソッドを用いて Web サーバに PD 送信する。HTTP は(2)プレゼンテーション、(3)セッション層の側面も持っているため、HTTP で定義する。

(4)トランスポート層:UDP

TCP は、3way handshake で接続を確立する。本

提案ではノードの数が大量にあり、1 コネクション確立に時間がかかることもあり、この時間で送れる PD も喪失してしまう。また、ノードから PD を CEP システムに送信できているのに、システムからの確認応答をノードが受信できず、応答を待ち続ける事象が発生し、そのノードが死んでしまう。さらに PD が CEP システムに正しく到着しなくても本アーキテクチャの動作に影響を与えない。

(5)ネットワーク層:IP

IP は、異なるネットワークを跨ぐパケットルーティングを行うルーティングプロトコルである。ネットワークセグメント内とそのセグメント外を区別なく通信することができるため、IP で定義する。

(6)データリンク層:MAC(媒体アクセス制御), RLC(無線リンク制御), PDCP(パケットデータ収束)

(7)物理層:PHY

本提案では、SMD の LTE 回線を使用することを仮定するため、LTE 回線の下位プロトコルを用いる。よって(6)データリンク、(7)物理層を上記で定義する。

7.2 実装環境

提案アーキテクチャにおけるサーバへの POST 送信機能の確認、性能評価を行うため、提案アーキテクチャに基づくプロトタイプを開発する(図 6)。また、プロトタイプの開発環境、実行環境を以下に示す(表 2)。

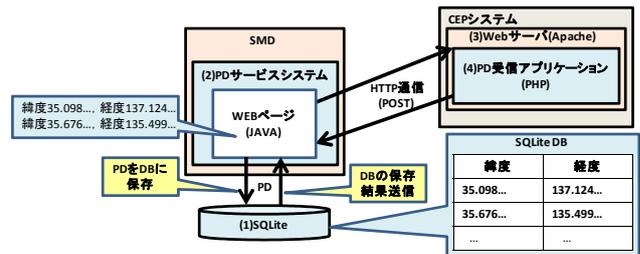


図6 プロトタイプの構成図

表 2 プロトタイプの開発環境と実行環境

	PDサービスシステム	外部システム
OS	Windows 7 SP1	Windows 7 SP1
CPU	Intel(R) Core(TM)2 Duo 2.53GHz	Intel(R) Core(TM)2 Duo 2.53GHz
メモリ	2.00GB	2.00GB
JDK	JDK 1.7.0_45	
Eclipse SDK	Eclipse SDK 4.3.1	
Apache		Apache 2.2.11
SQLite	SQLite 3.7.11	
AVD	Android 3.0	

以下にプロトタイプの仕様を示す。

(1) SQLite

プロトタイプでは、PD サービスシステムに打ち込まれた位置情報(緯度と経度)が保存されるデータベースである。

(2) PD サービスシステム

プロトタイプでは、JAVA で実装する。PD サービスシステムは、SQLite に位置情報(緯度と経度)の保存、削除、閲覧、SQLite に保存されている位置情報を Web サーバへの POST 送信する機能を持つ。

(3) Web サーバ

プロトタイプでは、Apache によりサーバを構築した。XAMPP Control Panel により start, stop の操作を行う。

(4) PD 受信アプリケーション

プロトタイプでは、PHP ファイルとして Web サーバの内部に存在する。PD サービスシステムから POST 送信されてきた位置情報(緯度と経度)を受信し、表示させる。

8 プロトタイプの評価

8.1 PD サービスシステムの POST 送信時間

プロトタイプの評価のため、プロトタイプに Calendar クラスを用いて送信開始時刻と送信完了時刻の取得をし、POST 送信時間の測定を行った(図 7)。

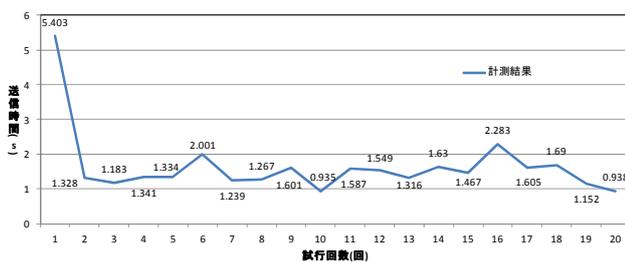


図 7 送信時間の計測結果

計測結果の 1 回目は、5.403s となっており 2-20 回までの値よりかなり時間がかかっている。これは、1 回目の送信時間の計測の際、Calendar クラスを呼び出しているためだと考えられる。結果より、安定した時間で PD を Web サーバに送信可能であることが確認できた。また送信時間計測を実施しなければすべての通信で、安定した送信時間で PD を送信可能であると考えられる。

8.2 UDP, TCP の初期ノードとの通信所要時間比較

TCP, UDP の評価のため、ネットワークシミュレータ NS-2 を用いて、初期ノードから目的ノードまでの到達時間を通信プロトコル AODV, DSR で計測した(図 8)。

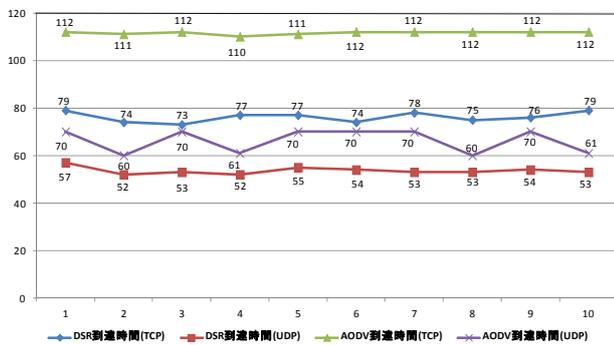


図 8 UDP, TCP の目的ノード到達時間(ms)

UDP の AODV の通信時間の平均が 66.2ms, DSR の通信時間の平均が 53.6ms, TCP の AODV の通信時間の平均が 111.6ms, DSR の通信時間の平均が 76.2ms であった。結果である各プロトコルの平均を表 3 に示す。

結果より、UDP のほうが、データを高速に送信することが可能であると確認できた。

表 3 TCP, UDP 平均到達時間の比較表(ms)

	TCP		UDP		通信プロトコル比較
	平均到達時間(ms)	標準偏差	平均到達時間(ms)	標準偏差	
AODV	111.6	0.66	66.2	4.66	UDP:約41%高速
DSR	76.2	2.04	53.6	1.43	UDP:約30%高速
ルーティングプロトコル比較	DSR:約32%高速		DSR:約19%高速		

表 3 より、TCP では DSR が約 32%高速であり、UDP では DSR が約 19%高速である。よって本提案では UDP を用いるため、DSR との組合せの効果が高いと思われる。

またトランスポート層のプロトコルによってルーティングプロトコルの到達時間が変化することを設計の際考慮する必要があると考える。

9 今後の課題

今後の課題として以下のことが挙げられる。

- (1)SSL 通信の実装
- (2)定義したプロトコル群の実装
- (3)ネットワークシミュレータではなく正式な SMD を用いた計測

10 まとめ

本稿では、VICS エリア外からリアルタイムに PD を送信し、アプリケーションが処理可能とするアーキテクチャを提案した。

MANET と CEP システムを連携させるために、CEP 入力アダプタのプロトコルを OSI 参照モデルに基づき定義した。これにより、VICS エリア外から交通情報を高速に送信可能であること、クラウド上にリアルタイムに PD を格納可能となることより、PD をリアルタイムに送信できる。

ネットワークシミュレータを用いて、TCP と UDP における目的ノードまでの PD 到達時間を測定し、アーキテクチャへの適合性を評価した。さらに、提案アーキテクチャを評価するため、HTTP POST メソッドを用いてデータを送信するプロトタイプを実装し、その妥当性を示した。

11 参考文献

- [1] 富士通株式会社, 先進技術でビッグデータ時代を勝ち抜く,第4回復合イベント処理の効果,2012, <http://special.nikkeibp.co.jp/as/201207/middleware/vol4.html>.
- [2] 本田技研工業, インターナビ・リンク プレミアムクラブとは, <http://www.honda.co.jp/internavi/about/>.
- [3] 板垣 朝子, フローティングカーデータを核に自車の安全・安心から社会貢献へ, 2011,<http://wirelesswire.jp/special/201108/01/article/1.html>.
- [4] 大畑 紀博, 無線マルチホップネットワークとインターネットとの統合技術に関する研究, 2006,<http://www.sk.tsukuba.ac.jp/SSE/degree/2005/thesis/200430110.pdf>.
- [5] F. Terroso-Sáenz, et al., Cooperative Approach to Traffic Congestion Detection With Complex Event Processing and VANET, IEEE Trans on ITS, Vol. 13, No. 2, 2012, pp. 914-921.